

OKTOBER 1984 ÖS 50,– /Sfr 6,– DM 6,–

Eine Markt & Technik Publikation

10|84 DAS MAGAZIN FÜR COMPUTER-FANS

Alles über Drucker und Plotter

★ 8 Tests ★ Marktübersicht
★ Die wichtigsten Hardcopy-Routinen

Floppy-Intensivkurs

Listing des Monats

Turbo-Lader für die 1541

Mit Akustikkoppler und Modem:

So nützt man Datex-P und ausländische Netze

Hackerclub:

»Wir sind keine Kriminellen«

So macht man Basic-Programme schneller

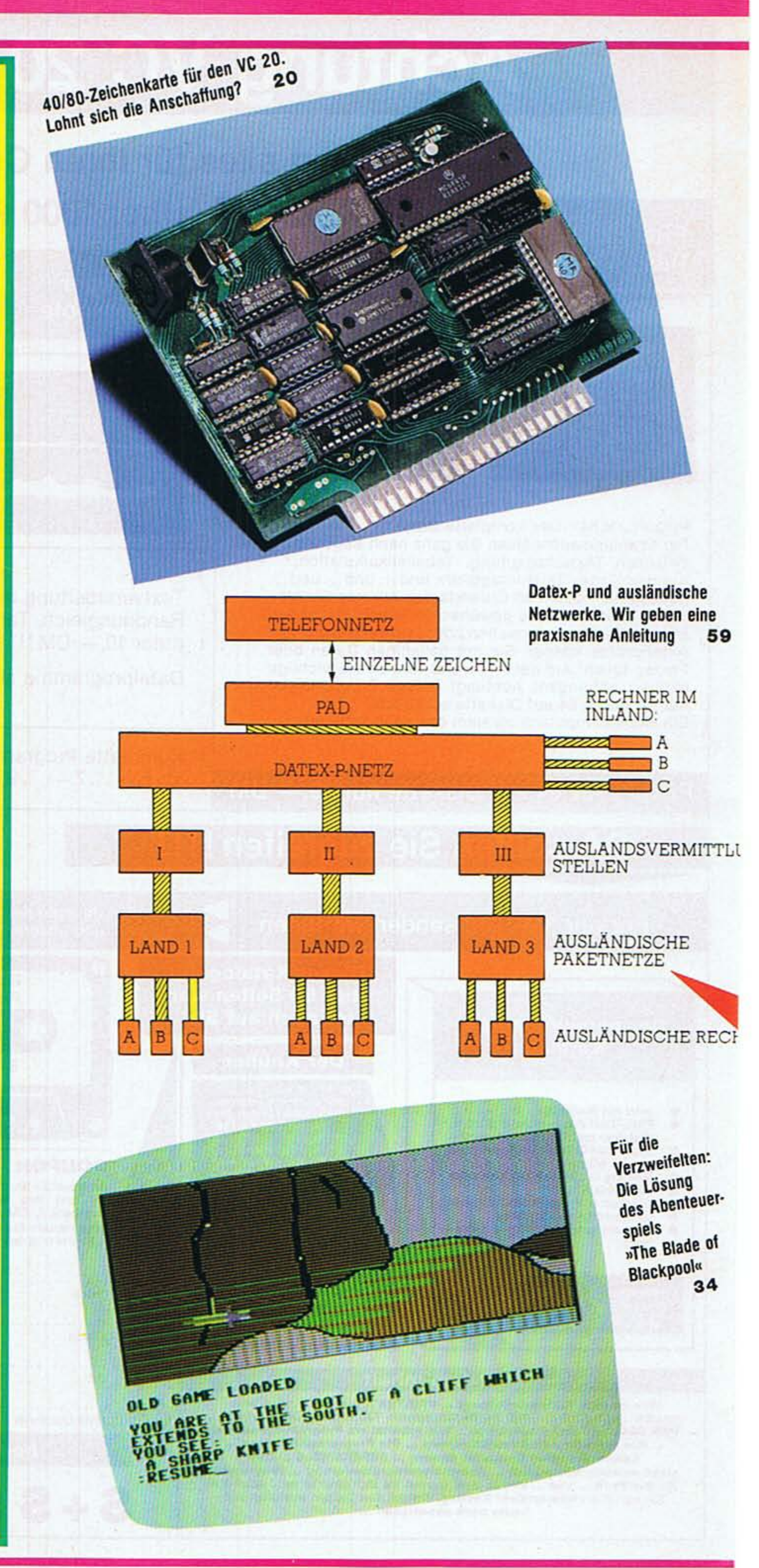
Software-Test:
VizaWrite und VizaStar

Das Top-Paket für den C 64

Bringt Profileistung beim Schreiben, Kalkulieren und Daten verwalten

Supercopy ★ Zehnertastatur am User-Port ★ VC 20-Kurs ★ Diskettenorganisation Farbige Hardcopy ★ Unterprogramm-Bibliothek ★ Test: 40/80-Zeichen-karte ★ Tips und Tricks für VC 20 und C 64

INHALT



40/80-Zeichenkarte für den VC 20. Lohnt sich die Anschaffung? 20

Datex-P und ausländische Netzwerke. Wir geben eine praxisnahe Anleitung 59

Für die Verzweifelten: Die Lösung des Abenteuerspiels »The Blade of Blackpool« 34



Es regt sich etwas auf dem Markt der alternativen Floppys **10**

Vom Typenrad- über Matrixdrucker zum Printer/Plotter. Sieben verschiedene Systeme im Test — mit Marktübersicht **22**

Es steckt noch einiges in der 1541. Hypra-Load, das Listing des Monats, lädt fast alle Programme bis zu fünfmal schneller **67**



Drucker oder Plotter?

Man braucht keinen teuren Farbdrucker, um eine farbige Hardcopy zu erstellen: es geht — wie wir mit einer in diesem Heft veröffentlichten Routine beweisen — auch mit dem preiswerten Printer/Plotter 1520. Dieses Listing zeigt zugleich, daß die Möglichkeiten eines Plotters häufig unterschätzt werden — er kann eben mehr, als nur schöne Linien und Kurven zeichnen. Er ist zwar nicht der schnellste, bringt aber auch Listings und Texte sauber zu (Rollen-)Papier. Bei der letzten Commodore-Fachausstellung demonstrierte ein Softwarehaus sogar, daß man Formulare unter Umständen leichter mit einem Plotter (der Formate wegen freilich nicht mit dem 1520) als mit einem üblichen Drucker ausfüllen kann. Ein bißchen Phantasie kann nicht schaden: Sicher gäbe es noch eine ganze Reihe weiterer Anwendungen für Plotter — auch die Flachbettplotter für DIN A4 oder gar DIN A3 sind ja nicht mehr unerschwinglich teuer.

Wir gehen in diesem und in den nächsten Heften ausführlich auf die wichtigsten Ausgabegeräte, ihre Stärken und Schwächen ein: Matrixdrucker, Typenraddrucker und Plotter. Ergänzend veröffentlichen wir im Listingteil häufig benötigte Routinen für die wichtigsten Geräte. Wir würden uns freuen, wenn sich Leser melden, die bessere Druckprogramme geschrieben haben, über interessante Anwendungen berichten oder Tips für die Lösung bestimmter Anschlußprobleme geben können.

Michael Pauly, Chefredakteur

Aktuell

Aktuelles aus der Datenfernübertragung

Commodore-64-Mailboxen in Krefeld

In Krefeld gibt es zwei Mailboxen. Beide laufen mit dem Automodem-Interface und sind vorwiegend abends zu erreichen. Die erste — von der Firma Hawischa Infosystems — hat die Nummer (02151) 779243. Dieses Mailbox-System ist hauptsächlich für den C 64-Anwender gedacht. Andere Systeme nehmen aber auch daran teil. Für den Anwender-Menüpunkt »Top-Secret«, wird ein Passwort verlangt. Dieses Passwort kann man sich auf Anforderung zuschicken lassen. Was geboten wird, ist bis auf weiteres unbekannt. Das andere Mailboxsystem wird von dem »Erfinder« des Automodems auf privater Basis betrieben und hat die Nummer (02151) 20130. Jeder, der ein Mailbox in Betrieb nimmt, kann auf Anfrage seine Adresse und Nummer veröffentlichen lassen.

Hackertreff in München

Ende Juli trafen sich 20 Münchener Hacker in einer Wirtschaft vor dem Sendlinger Tor. Bekannt gemacht wurde der Termin im Mailboxsystem »TEDAS« und in der RMI-Mailbox von der RMI-Nachrichtentechnik GmbH. Viele Teilnehmer waren aus den Mailboxen unter ihrem Pseudonym bekannt, wie zum Beispiel Professor Falken, König Ludwig, Kaiser Ferdinand, Donald Duck, Holy Hacker, Rastro Fodens, Graf Binaer Statement von Hohen Opcode, Christian Schwarz-Schilling. Die hauptsächliche Aktivität bestand natürlich im Tauschen von Telefonnummern, NUAs, NUIs, User-IDs und Passwörtern. Ferner wurden Adressen ausgetauscht und gefragt, wer welche Software für was hat. Zu erreichen sind die Münchener Hacker im RMI-Net noch unter dem User SENDLINGER TOR; diese Kontaktstelle soll aber bald aufgelöst werden. Dann kann ein Außenstehender die Mitglieder nur noch einzeln unter dem jeweiligen Pseudonym erreichen. Der Termin für das nächste Treffen, das wahrscheinlich am selben Ort stattfindet, steht noch nichts fest.

(Thomas Obermair/aa)

Hohe Gestaltungsqualität, hoher praktischer Nutzen und ergonomische Anpassung, technische und formale Eigenständigkeit und Gebrauchs-Visualisierung waren unter anderem die Wertmaßstäbe, die die zwölf if-Juroren und Sachverständigen ihrer Bewertung zugrunde legten. Nahezu dreihundert Unternehmen aus vierzehn Ländern beteiligten sich an dem Designer-Wettbewerb.

Commodore kann reich-



Modem aus Großbritannien

Dieses Modem aus England ist speziell für den Acorn BBC Computer konzipiert worden. Es arbeitet sowohl im 300/300 als auch im 1200/75 Baud-Modus und kann zwischen dem (europäischen) CCITT- oder dem (amerikanischen) Bell-Standard umgeschaltet werden. Der Anschluß an den BBC Model B Computer findet einfach über ein fünfpoliges DIN-Kabel statt. Natürlich hat das Gerät noch keine FTZ-Nummer. (aa)

Software Express GmbH, Hugo Viehoffstraße 84, 4000 Düsseldorf 30, Tel. 0211-424223

Interessante Modems

Um es vorwegzuschicken, die hier beschriebenen Modems wird es vorläufig nicht auf dem deutschen Markt geben, da ist die Bundespost vor. Dennoch sollen sie kurz vorgestellt werden, um vielleicht wenigstens als Denkanstoß für unsere gute alte BP dienen zu können.

Das Modem-Phone stellt eine Kombination aus einem Telefon und einem Modem dar. Der Telefonteil kann also wie ein normales Telefon (in den USA, also mit Bell 103 und CCITT V.21/V.23 Norm ausgestattet) gehandhabt werden. Der Modemteil läßt sich über eine RS232C-Schnittstelle mit allen Computern die selbige besitzen (beispielweise IBM, Apple, Radio Shack (TRS 80 etc.), DEC und anderen) betreiben. Mit einem entsprechenden Interface natürlich auch mit dem C 64. Das Modem kann sowohl im 300 als auch im 1200 Baud-Modus (asynchron) betrieben werden. Dieses Modem-Phone ist also hauptsächlich auf den amerikanischen (und englischen) Markt zu geschnitten. Deshalb die Frage an den Bundespost-Minister: Wieso gibt es so etwas noch nicht in Deutschland? Müssen wir eigentlich immer hinter den anderen herlaufen oder hat das (Kupfer-) Kabelprojekt wirklich einen so großen Stellenwert? (aa)

SX 64 ausgezeichnet

Eine internationale Jury hat Commodore für ihr vorbildliches Industriedesign ausgezeichnet. Sie erhielt für ihren seit dem Herbst '83 auf dem Markt befindlichen SX 64 (die tragbare Version des Commodore 64) das if-Emplem des Vereins »Die gute Industrieform Hannover e.V.«, das alljährlich zur Hannover-Messe verliehen wird.

lich stolz sein auf diese Auszeichnung. Dennoch gibt es aus unserer Sicht (und aus der des Lesers) auch einige Punkte die Anlaß zu Kritik geben. So kann man den zu klein geratetenen Monitor sicherlich nicht als sehr ergonomisch bezeichnen; genaueres ist in der Ausgabe 4 nachzulesen. Einige Leser monierten mit Recht, daß in Deutschland der SX 64 nur mit dem (zwar deutschen) Handbuch für den Commodore 64 ausgeliefert wird. Nützlicher wäre es da doch, den SX 64 mit seinem englischen Handbuch auszustatten. Dieses SX 64-Handbuch ist nämlich — gemessen an Commodore-Maßstäben — um einiges umfangreicher und besser als das zum Commodore 64. Der Idealzustand wäre natürlich eine gute deutsche Übersetzung. Die Anregung ist da, was Commodore macht, bleibt abzuwarten. Der SX 64 ist es sicherlich wert.

(aa)

Super Disk Drive

Nun gibt es sie auch in Deutschland, die MSD Super Disk II.

In der CES-Berichterstattung aus Chicago wurden sie schon einmal kurz angeschnitten, die Laufwerke von MSD Systems. Damals noch mit der Einschränkung, auf dem deutschen Markt müsse mit der Einführung wohl noch eine geraume Zeit gewartet werden. Diese Wartezeit ist nun vorbei. Softline tritt als Importeur auf.

Was bietet nun dieses Doppellaufwerk? Zunächst einmal zwei normale serielle Anschlüsse, wie wir sie auch von der VC 1541 her gewohnt sind. Zudem aber ist, über ein IEEE-Interface, auch der parallele IEC-Bus der größeren CBM-Systeme verfügbar. Durch den Einsatz dieses Busses ergeben sich natürlich erhebliche Geschwindigkeitsvorteile. So beträgt die Zeit, um eine volle Diskette zu kopieren 1 Minute und 55 Sekunden. Darin sind die Zeiten für das Formatieren der neuen, leeren Diskette, das Kopieren und das Verifizieren eingeschlossen. Die reine Formatierungszeit für eine neue Diskette beträgt 17 Sekunden. Diese Werte beziehen sich nur auf den parallelen Bus. Arbeitet man mit dem seriellen Anschluß, unterscheiden sich die Zeiten nicht wesentlich von denen der VC 1541. Dieser Anschluß ist aber dennoch nötig: denn nicht alle Programme laufen mit der parallelen Schnittstelle. So hatten wir beispielsweise Schwierigkeiten mit kopiergeschützten Programmen. Die Aussage der 100prozentigen Kompatibilität kann also noch nicht gemacht werden. (aa)

Info: Softline, R. Alverdes, Schwarzwaldstraße 8a, 7602 Oberkirch, Tel. 07802-3707

Das Klavier am C 64

Wersiboard Music 64

Ein neues Musiksystem für den Commodore 64 hat die Firma Wersi vorgestellt. Es nennt sich Wersiboard Music 64 und besteht aus einem Keyboard mit 49 Tasten, einem Interface sowie der zugehörigen Software.

Somit gehören nun fingerverrenkende Spielübungen auf der Schreibmaschinentastatur des Commodore 64 der Vergangenheit an. Musiker und musikliebende Computerfans können jetzt Melodien und bis zu dreistimmige Lieder in gewohnter Weise auf einer Klaviatur spielen. Die Computertastatur benötigt man nur noch zur Bedienung der Software. Jeder auf der Klaviatur angeschlagene Ton klingt sofort aus dem Monitorlautsprecher. Auf dem System läßt sich also spielen wie auf einem kleinen dreistimmigen Synthesizer beziehungsweise einer dreistimmigen Orgel. Zur Klangerzeugung dient der SID-Chip des C64. Die Software wird auf Diskette geliefert und besteht aus den 2 Programmen »Mono 64« und »Poly 64«.

Mono 64 ist für einen monophon spielbaren Synthesizer. Monophon bedeutet einstimmig. Mit diesem Programm lassen sich also nur Melodien spielen. Dafür bietet es umfangreiche Klangeinstellmöglichkeiten. 13 Klangprogramme stehen während des Spieles zur Auswahl. Ein Set von 13 Demonstrations-Klängen befindet sich bereits auf der Diskette: Trompete, Blechbläser, Klarinette, Glokken, Flöte, Gitarre, Streicher, Klavier, Orgel und Akkordeon gehören hierzu. Sie klingen alle sehr echt. Ist man mit diesen Klängen nicht zufrieden oder will man neue »erfinden«, sind Experimenten kaum Grenzen gesetzt. Nahezu alle Parameter des SID-Chips können in weiten Bereichen verändert werden. Attack, Decay, Sustain und Release lassen sich getrennt regeln. An Filter-Modi stehen Tiefpass, Hochpass und Bandpass zur Verfügung. Einer der drei SID-Oszillatoren ist als langsamschwingender Modulationsoszillator geschalten. Mit diesem lassen sich Tremolos und Vibratos erzeugen. Die beiden anderen Oszillatoren spielen die Melodie. Sie lassen sich in der Tonhöhe gegeneinander verstimmen und mit unterschiedlichen Kurvenformen ausstatten, wodurch interessante Klangvariationen möglich sind.



Poly 64 ist das Programm für einen polyphonen Synthesizer. Maximal sind natürlich nur drei Stimmen gleichzeitig spielbar. Dieses Programm weist etwas weniger Klangvariationsmöglichkeiten als das vorhergehende auf. Ein Modulationsoszillator ist nicht vorhanden. Über das Menü kann man nur 5 Klangfarben auswählen. Spinett, Accordeon, Flöte, Banjo und Tubular Bell sind schon fertig programmiert. Über die Funktionstasten des Commodore 64 lassen sich Attack, Decay, Sustain, Release, Lautstärke, Filterfrequenz, Kurvenform der Oszillatoren und die Pulsbreite ändern. Leider können beide Programme gespielte Melodien und Lieder weder in Notenschrift am Bildschirm ausgeben, noch speichern (Sequenzer). Ergänzungs-Software mit diesen Möglichkeiten wäre sehr wünschenswert. Hoffentlich nehmen sich die Softwareingenieure der Firma Wersi dieser Probleme bald an.

(Richard Aicher/aa)

Info: Wersi, Industriestraße, 5401 Halsenbach/Hunsrück, Tel. 06747-7131

Die Alternativen kommen

Es rührt sich was auf dem Markt der alternativen Diskettenlaufwerke.

Knock out für die 1541?

Nicht nur auf dem amerikanischen Markt tut sich einiges in bezug auf die alternativen Laufwerke für die VC 1541. So haben sich einige Importeure gefunden, die Laufwerke aus Hongkong oder Korea anbieten, teilweise aber auch aus europäischer Produktion.

Da wäre zunächst einmal das Doppellaufwerk Micropower 2000 von WCS (Wemper Soft-Comp) in Aumühle. Dieses Laufwerk bietet ebenfalls wie das MSD Super Disk zwei serielle Anschlüsse und einen Anschluß über den IEEE-488-Bus. Über den zuletzt genannten Bus ist das Micropower 2000 an die größeren CBM-Computer anschließbar. Es versteht die Befehle von Basic 3.0 und 4.0.

Beispiel: Basic V 3.0; LOAD"0: Name",8 oder LOAD"1: Name",8 zum Laden eines Programms von Diskette in Laufwerk 0 beziehungsweise Laufwerk 1. In Basic V 4.0 sehen die entsprechenden Befehle so aus; DLOAD "Name",D0 oder DLOAD "Name",D1. Eine genaue Angabe wie sich die Micropower 2000 mit dem C 64 verträgt kann noch nicht gegeben werden. Ein ausführlicher Testbericht wird aber noch folgen.

Info: WCS Wemper Comp-Soft, Börnsener Str. 6, 2055 Aumühle/Hbg., Tel. 04104-5365



Links das Doppel-Laufwerk Micropower 2000.

Oben das Wafertape-Laufwerk mit dem Endlosband. Unten das Einzellaufwerk für den VC 20/C 64.

Alternative 1541

Von ncs Computersysteme in Nettetal wird ein vollkommen kompatibles Laufwerk zur 1541 angeboten. Das heißt auf den IEC-Bus wurde verzichtet. Mit diesem Laufwerk hat man also eine 1541 vor sich, nur mit wesentlich verbesserten Leistungsmerkmalen. Diese beziehen sich allerdings nicht auf die Geschwindigkeit, sondern nur auf die Zuverlässigkeit. Die 100prozentige Kompatibilität bedingt aber auch, daß das Commmodore-Floppy-Betriebssystem 1:1 übernommen werden müßte. Und da liegt der Hase im Pfeffer. Commodore läßt die Freigabe dieses Laufwerkes nicht zu. In der letzten Ausgabe hatten wir über ein ähnliches Laufwerk von WM-Computer berichtet. Bei diesem Laufwerk liegt dieselbe Sachlage vor. Das Laufwerk ist vorhanden, es kann geliefert werden, nur es darf nicht. Das Copyright des Floppy-DOS liegt selbstverständlich bei Commodore, und die geben es nicht frei.

Die Importeure dieser beiden Laufwerke bemühen sich nun, ein anderes Disketten-Operation-System in ihre Laufwerke einbauen zu lassen. Eine traurige Tatsache, da gäbe es nun zwei zuverlässige Laufwerke auf dem Markt, doch sie dürfen nicht verkauft werden.

Wir halten Sie über den Stand der Dinge auf dem laufenden.

Wafertape

Ebenfalls von ncs Computersysteme wird es demnächst ein Wafertape-Laufwerk für den C 64 geben. Die Übertragungsrate liegt bei 19,2 KBit pro Sekunde. Das SF1403 genannte Wafertape-Laufwerk kann direkt an den C 64 oder den VC 20 angeschlossen werden, und bietet aber außerdem noch eine Centronics- sowie eine RS232-Schnittstelle. Die Datenspeicherung findet auf einem sogenannten magnetischem Endlosband (in etwa einer verkleinerten Musikkassette vergleichbar) statt. Es können je nach Bandlänge bis zu 140 KByte an Daten gespeichert werden. Die Handhabung entspricht der einer Diskette, also mit Directory, sequentiellen Dateien etc. Das Doppel-Laufwerk soll 599 Mark kosten, und wird wahrscheinlich Ende dieses Jahres erhältlich sein. (aa)

Info: ncs Computersysteme GmbH, Klemensstraße 7, 4054 Nettetal 2, Tel. 02157-1616

KREATIVES CHAOS

Der Computer bietet in der Datenfernübertragung ungeahnte Möglichkeiten. Aber die Gefahr des Mißbrauchs ist gegeben. Hier werden immer wieder die Hacker genannt. Ihre bekannteste »Vereinigung« ist in Deutschland der Chaos Computer Club in Hamburg. Das folgende Interview ist weder eine Rechfertigung noch eine Verurteilung der Hacker. Der Artikel soll einen Einblick in die »Philosophie« des CCC-Teams bieten.

Dieses Interview wurde telefonisch durchgeführt, wobei die Interviewten, zwei Mitglieder des CCC, vor einem Telefonverstärker saßen. Das wohl wichtigste Ergebnis ist: Das Bild vom Hacker, der nur aus Spaß und aus Freude an der Sache selber in Datenbanken eindringt, ist nicht mehr aktuell. Die »Hacker« vom Chaos Computer Club (CCC) verstehen sich selber als Kommunikationsexperten, die das neue Medium Datenfernübertragung per Modem zur weltweiten Kommunikation nutzen. Letztendlich soll jeder unzensiert Informationen austauschen können. Das langfristige Ziel ist die Installation von besseren, objektiveren Möglichkeiten zur freien Meinungsbildung. Ein »Hacker« dieser Couleur sieht also in seinem Handeln eine gesellschaftliche Aufgabe.

Was macht der Chaos Computer Club überhaupt?

Wir verbreiten Informationen über neue Medien und unsere Erfahrungen und tauschen Informationen über die verschiedensten Sachen aus. Computer sind so etwas wie ein neues Medium für uns und diese Datenverbindungen sind für uns eine neue Form von Straßen und öffentlichen Plätzen, auf denen wir uns bewegen. Wir möchten mit der Zeitschrift »Datenschleuder« andere Leute auf die bestehenden Möglichkeiten und die Gefahren hinweisen.

Das ist doch ein bißchen harmlos ausgedrückt, oder?

Ich weiß nicht, was Du willst. Die ganzen Zeitschriften oder was immer so berichtet, die versuchen uns immer in eine leicht kriminelle Ecke hineinzudrücken. Das ist überhaupt nicht unser Sinn und Zweck oder gar unser Anliegen. Wir möchten ganz einfach nur miteinander Kontakt halten und aufnehmen und natürlich auch — also ich persönlich — in den Rechnern gucken, was da so abläuft. Um da reinzukommen muß man ein bißchen kreativ sein.

Wie kam es zur Gründung des Chaos Computer Clubs?

Vor drei Jahren haben sich ein paar Leute zusammengehockt, und gemeint, daß man mit Computern auch andere Sachen machen kann, als das, was zumindest hier in Deutschland zu diesen Zeiten üblich war. Da ist dann eine Kleinanzeige in einer Tageszeitung erschienen, und in Berlin fand ein Treffen von zwei Dutzend Leuten aus dem Bundesgebiet statt. Da haben wir damals beschlossen in Kontakt zu bleiben und Erfahrungen auszutauschen. Die Idee war einfach die, eine Zeitschrift zu machen, aber zu dem Zeitpunkt war das alles zu locker, und es hat nicht hingehauen. Weiter haben sich Kontakte zu den USA ergeben, zu Cheshire Catalyst (dem »König« der Hacker in den USA, d.Red.) und zu dessen Zeitschrift Cap, die eine ganze Reihe von Leuten hier im Bundesgebiet schon kannten und abonniert haben. Auf der TELECOM im Herbst 83 habe ich ihn selber kennengelernt und darüber eine Doppelseite in einer Tageszeitung geschrieben. Und diese Doppelseite hat eine ganze Menge an Resonanz hervorgebracht. Zur Jahreswende war es dann soweit, eine Zeitschrift zu machen. Vorher gab es die Idee, alle Informationen über Floppy auszutauschen, aber das war alles inkompatibel und hatte keinen Zweck. Die Zeitschrift wurde zur Jahreswende angekündigt und brachte binnen einer Woche an die 100 Zuschriften. Da war das hier in Hamburg einfach klar: Wenn 100 Leute das haben wollen, muß die einfach gemacht werden. Im März war es dann endlich soweit, fertig war die erste Nummer. Als dann die erste und die zweite Nummer heraus waren, ergab sich eine wahre Briefflut. Wenn ich meine beiden Hände hochkant hinhalte, das entspricht der eingegangenen Post.

Wieviele Abonnenten hat die Zeitschrift?

Wir sind einfach den Weg gegangen, daß der, der geschrieben hat, auch Informationen bekam, egal ob er jetzt Geld schickte oder nicht. Die Zahl der Abonnenten bewegt sich bei etwas über 200.

Wie schauen jetzt eure finanziellen Möglichkeiten aus?

Schlecht! Die Produktionskosten von der Zeitung liegen pro Exemplar rund bei 10 Pfennig, die Doppelseite DIN-A-3 bedruckt. Auf der anderen Seite ist das Teuerste halt das Porto. 50 Pfennig für jedes Ding. Auf ein Exemplar kommen drei Exemplare, die nicht bezahlt sind, die wir einfach wegschicken, sei es zur Probe, oder auf eine Anfrage, wo nichts bei lag. Sie finanziert sich eben gerade über der Grasnarbe, aber wir haben die Hoffnung, daß eine Reihe von Leuten dazu bereit ist, dafür auch zu zahlen. Wenn Geld reinkommt, kann man auch weitermachen.

Und von den 200 Abonnenten hat jeder etwa 30 Mark gezahlt?

So ungefähr. Die sind auf den Faktor 1 gekommen, ein paar haben auch noch etwas gespendet, aber das war leider weniger als erhofft. Die Herstellung einer Datenschleuder kostet ungefähr 1000 Mark pro Ausgabe.

Wie lange bekommt einer die DS für 30 Mark zugeschickt?

Ein Chaos-Jahr. Das ist effektiv ungefähr ein Jahr, mit ungefähr 10 Ausgaben. Da müßt Ihr unbedingt unsere Adresse angeben, sonst rennen sie Euch die Bude ein.

Wieviele feste Mitglieder sind jetzt dabei?

Zur Mitgliederstruktur: Das ist ziemlich offen und frei. Jeder, der Informationen für uns hat, der trägt sie einfach bei, ob er nun Abonnent oder nur Bezieher der Datenschleuder ist, oder ob er

gar nichts von der Datenschleuder weiß. Es gibt Leute, die beschäftigen sich mit dem Modembauplan, wir stellen zum Beispiel Platinen her, vertreiben sie auch. Dann machen wir die Datenschleuder, und haben die Kommunikation über Computer. Das ist einfach eine offene Struktur, wo jeder, der Lust hat, mitmachen kann und es irgendwann bleiben läßt. Wir sind aber am Überlegen, ob wir das Ganze in Vereinsstrukturen einbinden sollen. Es gibt ein paar Stellen nach außen, wo wir einfach auf Probleme stoßen. Aber auf der anderen Seite gefällt uns einfach diese offene Struktur. Das ist ein Dilemma.

Eine ungefähre Zahlenangabe läßt sich doch machen, oder?

Wenn man das auf der Ebene der Abonnenten ansetzt, kommt man so auf über 200, davon etwa 30 in Hamburg. Das sind immer so fließende Grenzen. Soll ich jetzt einen Grafiker, der mal ein paar Bilder für die Datenschleuder malt als Mitglied einstufen oder nicht? Das sehen wir nicht so eng. Da kommt jemand dazu, findet das interessant, macht dann einfach mit. Das ist auch so bei den Treffen, die wir einmal in der Woche in einer Kneipe hier in Hamburg machen. Da kommen wechselweise immer andere vorbei, ein Stammtisch eben. Zum anderen treffen wir uns einmal im Monat allgemein zum Informationsaustausch.

Welche Leute machen mit?

Das geht vom Piloten über einen Schlosser, Leute mit einem eigenen Laden bis hin zu Arbeitslosen und Studenten. Sie sind zwischen 16 und 35 Jahre alt. Nicht älter.

Wie beurteilt Ihr die rechtliche Lage Eurer Aktivitäten?

Wir mögen es überhaupt nicht, wenn wir in irgendeine Ecke gedrängt werden. Jetzt wird gesagt, alles seien Kriminelle, die Datenbanken knacken und einen großen Unfug treiben. Wir erheben grundsätzlich nicht den Anspruch, daß wir uns an alle Gesetze und Regeln halten, zum Beispiel bezogen auf die Verwendung von nicht FTZ-geprüftem Gerät. Wir wollen die Bundespost davon überzeugen, daß das wie in England gehandhabt wird, also grob gesagt, die Nutzung von nicht FTZ-geprüftem Gerät zugelassen wird. Das ist eine klare Forderung von uns. Wir sind das Gegenteil von Computerkriminellen, die wegen des eigenen finanziellen Vorteils in Computersysteme eindringen und irgendwelche Sachen von dort verkaufen; genauso wie wir uns ganz klar von Leuten distanzieren, die Software kopieren und dann weiterverkaufen. Auf der anderen Seite ist das bei uns so wie beim Patentrecht: Wenn wir bestimmte Dinge ausschließlich für den privaten Gebrauch benutzen, kann es durchaus vorkommen, daß wir Geräte verwenden oder Sachen machen, die irgendwie nicht zugelassen sind.

Was sind die wichtigsten Ziele des CCC?

Eine ganz wichtige Zielsetzung ist das neue Menschenrecht auf weltweiten, freien Informationsaustausch. Ungehindert. Das ist eine Chance, die die elektronischen Medien einfach bieten. Es passiert in einigen extremen Fällen, sagen wir mal bei Telefonaten mit Israel, daß sich die Zensur einschaltet und die Verbindung abbricht. Und in die Sowjetunion gibt es überhaupt keinen Selbstwählverkehr. Sonst ist das Telefon ja ein Hilfsmittel, um mit Menschen in aller Welt Verbindung zu bekommen und unzensiert zu reden. Und das ist ein ungeheurer Fortschritt, wenn man 200 Jahre zurückdenkt. Und diese Entwicklung wollen wir in Richtung auf die neuen Medien weitertreiben; wir versuchen einfach, die internationale freie Diskussion zu fördern, also so etwas wie der »freedom of information act« in den USA erreichen, weltweit.

Versteht Ihr jetzt unter frei auch kostenlos?

Das wäre schön. Aber angesichts meiner Telefonrechnung würde ich sagen, daß das schon lange nicht mehr kostenlos ist. Das ist eine Wunschvorstellung, die sich so nicht verwirklichen läßt. Es ist aber viel billiger als zum Beispiel jetzt dauernd herumreisen zu müssen; es ist so möglich, daß zu relativ geringen Kosten internationale Konferenzen stattfinden. Und an dem Punkt stinken wir natürlich auch gegen eine restriktive Gebührenpolitik der Bundespost an. Es ist um den Faktor 4 billiger, von den USA nach hier zu telefonieren als von hier in die USA. In den USA sind Ortsgespräche zum Beispiel kostenlos. Die Bundespost hat ihr Monopol, und sie nutzt es aus. So ist es in den USA möglich, als Funkamateur das Funknetz mit dem Telefonnetz zu koppeln, was in der Bundesrepublik verboten ist. An solchen Stellen haben wir, einmal vorsichtig ausgedrückt, reformerische Vorstellungen.

Seht Ihr in dem Buch »Der Schockwellenreiter« von John Brunner irgendwelche Parallelen in Eurer Arbeit?

Das ist ein Buch, das eine Reihe von Leuten ziemlich beeindruckt hat. Das ist eine Aktualisierung von »Brave New World« oder »1984« auf elektronische Medien. Es schildert eine Reihe von Gefahren und Chancen, die zu wenig in der Diskussion sind. Da finde ich einfach, das ist ein richtiges Buch.

Wie, meint Ihr, wird sich das Ganze in den nächsten 10 Jahren weiterentwickeln?

Einrichtung von mehr Mailboxen. Die ganze Computerei wird das Miteinander der Menschen ganz schön beeinflussen. In viel stärkerem Maße als das Telefon. Als negatives Bild: Vor 10 Jahren gab es die ersten Videogruppen: »Neues Medium, kann man interessante Sachen damit machen, zum Beispiel eine Stadtteilvideo oder Betroffenenvideo.« Also ein Medium für Einfälle. Und was ist nach 10 Jahren herausgekommen? Ein absolutes Massending, mit Horror und Porno. Und in ähnlicher Weise sehe ich das im Negativen für die Computerei. Sie führt zu einer neuen Form von Orientierung auf die Maschine und Sprachlosigkeit. Ich sage nur »1926 Metropolis« als Stichwort. Wir wollen versuchen, die Leute von ihren Daddelspielen wegzuziehen und zu einem kreativeren Umgang mit dem Medium zu bewegen. Unsere Hoffnung ist, daß der Computer als neues Medium positiv zur Verständigung beiträgt.

Wie seht Ihr in diesem Zusammenhang die Zukunft für den CCC?

Wichtig ist, daß das gedruckte Medium nur eine Krücke und ein Übergangsmedium ist, womit alle angesprochen werden, die keinen Computer und kein Modem haben. Also für die, die nicht »online« sind. Auf Dauer wird eine gedruckte Sache an Bedeutung verlieren. Die Ecken in den Mailboxen, wo Neuigkeiten drinstehen, sind viel aktueller und interessanter. Was in der Datenschleuder steht, ist oft total veraltet, wenn sie rauskommt. Jene, die sich ein bißchen in Mailboxen herumtun, sind vom Informationsstand einfach vier Wochen weiter. Deshalb müßte sich der Informationsaustausch viel mehr auf die elektronischen Medien verlagern. Auch bei uns.

Was würdet Ihr jemandem raten, der sich für die Kommunikation als solche interessiert?

Online! Rein in die Dinger! Gucken und sehen, was ihm gefällt, ob er irgendwas findet, womit er etwas anfangen kann und was seinen Interessen entspricht. Und wenn er nichts findet, sollte er zusehen, daß er etwas aufbaut. Auf alle Fälle immer aktiv sein.

Wer da ein bißchen ernsthaft herangeht, wird rela-

Fortsetzung auf Seite 176



Steuercodes oder Interface?

Ich besitze die elektronische Schreibmaschine SILVER-REED EX 44 und das dazugehörige Interface mit der Bezeichnung SRI IF-UI84 SIGMA 8300 SP zum Commodore 64. Diesem Interface ist leider nur eine äußerst dürftige Beschreibung beigefügt, die in keiner Weise den Möglichkeiten entspricht. Einen Teil der Codes, die — entgegen Herstellerangaben — auch funktionieren, habe ich schon herausgefunden. Für mich sehr interessant wäre eine Umschaltmöglichkeit (elektronisch) der Zeilendichte (Pitch), um für Diagramme eine höhere Auflösung erzielen zu können. Daher meine Frage: Wer hat den Code dieses Interfaces geknackt und/oder kann mir helfen? Das Problem scheint die Belegung der ESC-Funktion mit 1/2-Space zu sein.

Eckehard Ledig

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines gutes Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen beziehungsweise Produkten. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene — dann schreiben Sie uns doch. Die Antworten werden wir in einer der nächsten Ausgaben publizieren. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

Was steckt in der Demodiskette?

*Als ich mir vor ein paar Monaten zu meinem Commodore 64 das Diskettenlaufwerk kaufte, lagen da zwei Disketten dabei. Die eine Diskette nennt sich »TEST/DEMO«, wo auch das DOS 5.1 drauf ist. Die andere Diskette, die sich »DEMODISKETTE« nennt, sind zahlreiche Demonstrationen drauf wie: Musikdemos, Grafikdemos, ein paar Anwendungen, Spiele und zu guter Letzt auch Werbung für den C 64. Auf dieser Diskette ist auch das Programm, wo ich Sie nun um eine Auskunft bitte. Das Programm besteht eigentlich aus zwei Programmen. Das erste Programm (der Lader), das sich »3000.EMU.BOOT« nennt, lädt, nach dem man es gestartet hat, das eigentliche Programm, das sich »EMULATOR« nennt. Das Programm »EMULATOR« wird vom Lader aus mit SYS 12*4096 gestartet. Man erhält dann folgenden Bildschirmaufbau:*

PET EMULATOR V0.5/081382
WITH
DOS MANAGER V5.1/071382
BY BOB FAIRBAIRN
(C) 1982 COMMODORE BUSINESS MACHINES
PET MODE
EMULATOR ON
READY

Und nun meine Fragen zu diesem Programm: Was kann man mit diesem Programm alles anfangen? Wie kann man es sinnvoll anwenden? Das einzige, was ich bisher herausgefunden habe, ist folgendes:
1. Das Programm »EMULATOR« liegt im Adressenbereich 49152-53247, also ist es zirka 4 KByte lang.
2. Fragt man den freien Speicher ab, mit PRINT FRE(0), so erhält man die Antwort : 31741. Es gehen also zirka 7 KByte verloren. Was ist nun mit diesen 7 KByte passiert?
3. Das einzige, das bei der Diskette beilag, ist folgender Text: »Hilfsprogramm, um Programme von CBM 3000 auf den C 64 zu verwenden«. Bezieht sich dieser Text nur auf Basic- oder auch auf Maschinenprogramme.

Ich bin sicher, daß ein großer Teil aller C 64-Besitzer, die sich eine Floppy zugelegt haben, dies interessiert, um die große Software-Vielfalt der CBMs für den C 64 auszunutzen.

Wolfgang Joachim

Textverarbeitung mit Seikosha GP-700A?

Obwohl ich einige Textverarbeitungsprogramme in meiner Sammlung habe, ist es mir noch nicht gelungen, meinen Text durch Einfügen von Formatierungsbefehlen farbig zu gestalten. Der Drucker wird derzeit mit einer Centronics-Schnittstelle der Marke Wiesemann Typ 9200 angesteuert. Mit welchen Textprogrammen kann ich farbig ausdrucken und welche Centronics-Schnittstellen muß ich an den Drucker anschließen?

Conny Scharfenberg

C 64 extern anhalten?

Wie kann man den C 64 über externe Beschaltung (nur Hardware) beliebig (über den DMA-Eingang vielleicht?) anhalten und wieder starten?

Ulrich Lang

Vielsaitig für C 64?

Ich besitze ein Programmpaket für den VC 20 mit Namen »Vielsaitig« von Commodore. Es handelt sich dabei um ein Programm zur Lösung musiktheoretischer Probleme. Es kann insgesamt 25 internationale Tonskalen in Tabulatur und Klaviatur grafisch ausgeben, Gitarrengriffe bestimmen oder die richtigen finden und vieles mehr. Da ich nun auf den Commodore 64 umsteigen möchte würde mich interessieren, ob es ein Programm für den C 64 gibt, das über die obengenannten Funktionen verfügt und eventuell mit besserer Grafik und besserem Eingabekomfort aufwarten kann?

Oliver Kreuzahler

Der C 64 als Effektgerät?

Wer hat Programme für den C 64 zur Benutzung als Effektgerät über den Audioein- und -ausgang am User-Port (Pin 3/5)? Dabei denke ich an Anwendungen wie Echo/Phaser/Flanger/Chorus und Verzerrer oder auch Oktaver und Vocoder. Ferner würde mich interessieren, wie der Audioeingang intern verschaltet ist und ob er einen A-D-/D-A-Wandler besitzt?

Patrick Bandenberger

Wo gibt es Testprogramme?

1) Gibt es ein »Testprogramm«, das alle Funktionen des Computers testet? (zum Beispiel durch User-Port, Baud-Rate und so weiter).
2) Wo kann man »Bauteile« für den C 64 kaufen (zum Beispiel 6526)?
3) Exbasic Level II wurde auch schon angesprochen. Mit DUMP erhält man die Werte der Variablen (aber nur die absoluten!). Wie kann man die Vorzeichen sichtbar machen? Ist es eine Macke vom Exbasic?
4) Was ist mit den Unterschieden der einzelnen »C 64-Jahrgänge«?

Peter Strempel

3032-Monitor am C 64?

Ich möchte den Monitor eines CBM 3032 an den C 64 anschließen. Ist eine Schaltung bekannt, die die von diesem Monitor benötigten Signale (Video, horiz. sync., verf. sync.) aus dem F-BAS-Signal des C 64 erzeugt?

Wolfgang Stumm

Simons Basic und DOS 5.1 gleichzeitig?

Die gleichzeitige Verwendung von Simons Basic und DOS 5.1 auf der Gerätekombination C 64 und VC 1541 funktioniert nicht. Obwohl Simons Basic einige Disk-Befehle anbietet, empfiehlt es sich, um jederzeit den Fehlerkanal abfragen zu können, das DOS 5.1 mit zu verwenden. Welche Abhilfe gibt es, um beides störungsfrei nebeneinander betreiben zu können? Wer bietet das Textverarbeitungsprogramm »VIZAWRITE 64« an?

Detlev Preisler

Textomat und Steuerzeichen?

Wer hat Erfahrung mit dem Programm »Textomat« und »Datamat« (neue Version) von Data Becker in Verbindung mit dem Görlitz-Interface? Ich wäre an einer Druckertabelle und den Steuerzeichen interessiert.

Axel Birtel

C 64-Platine kaufen?

Kann man bei Commodore oder woanders auch nur eine Platine des C 64 kaufen?
Eric Kratzin

C 64-Tastatur kaufen?

Woher bekomme ich eine Tastatur für den C 64?
Oliver Varoß

Hardcopy für Printer 4022?

Ich besitze einen C 64 mit IEEE 488-Interface und den Tractor-Printer 4022.
1) Ist ein Hardcopy möglich? (Hardcopy der IEEE 488 funktioniert nicht!)
2) Wie komme ich zur Sperrschrift (breiterer Ausdruck von Zeichen) beim 4022?
3) Wo liegt der Grafikspeicher der IEEE 448?
Gernot Zelzer

Horizontales Scrolling?

Wie kann man beim C 64 ein horizontales Fine-Scrolling (Punkt für Punkt) auf der Textseite simulieren? Mit welcher Speicherzelle kann man die Textseite Punkt für Punkt verschieben?
Christoph Bergmann

Geldfrage?

1. Bei welchen Disketten ist das Preis-/Leistungsverhältnis am günstigsten? (in bezug auf die Commodore-Floppy VC 1541).
2. Wie kann man Sprites drehen?
3. Bei welchen Geschäften/Versandhäusern bekommt man die billigste Hard-/Software?
4. Gibt es Möglichkeiten, mit dem Homecomputer Geld zu verdienen?
Hendrik Richter

Fortran auf dem C 64?

Ich besitze einen Commodore 64. Gibt es für dieses Gerät einen Fortran-IV-Compiler oder eine andere Möglichkeit, auf dem C 64 mit Fortran IV zu arbeiten?
Hergen Eilers

Der C 64 an der Stereoanlage?

Ich habe folgende Fragen:
— Wie schließe ich meinen Computer (C 64) an eine Stereoanlage an (Mono-Umschalter nicht vorhanden)? Es soll kein Stereoton simuliert werden.
— Wie kann ich die Erweiterung »Turbo-Tape« von Maschinensprache aus nutzen? (Wo sind die Einsprünge; wie werden die Parameter übergeben?) Kann man mit »Turbo-Tape« absolut laden und von Maschinensprache aus speichern?
— Kann man Maschinenprogramme im Diskettenpuffer ablegen und dort laufen lassen? Wie kann man sie starten, wenn das oben genannte möglich ist?
Thomas Denner

SM-Text 64 mit Umlauten?

Ist es möglich, mit SM-Text 64 mittels C 64 und Drucker MPS 801 auch Umlaute zu drucken? Wer kennt gegebenenfalls ein Hilfsprogramm?
Ulrich Uphagen

Aztec Tomb

Wer kann mir bei der Pflanze weiterhelfen?
Ausgabe: 8/84
Stefan Sache

So würden Sie nicht weiterkommen, denn die folgenden Schritte fehlen Ihnen noch:
— im Hausflur:
Eingabe: examine hallstand
Comp.: I can see empty jam jar
Eingabe: take jar
Comp.: O.K.
— beim Teich:
Eingabe: examine pool
Comp.: I can see fish-pool
Eingabe: catch fish
Comp.: It's now in jam jar
— Waste Ground:
Eingabe: examine fish
Comp.: It's magical it says: Plants need water and vanishes
Eingabe (2mal): fill jar — empty jar

Die Pflanze wächst und Sie gelangen mit »climb beanstork« in die nächste Abenteuerstufe.
Olaf Amblank

Probleme mit Speichererweiterung

3-KByte-Spiele funktionieren nicht mit dem 16-KByte-Modul?
Ausgabe: 7/84
Daniel Hüller

Meine Antwort haben Sie offenbar falsch verstanden. Um beim VC 20 eine 3-KByte-RAM-Erweiterung zu simulieren, benötigt man lediglich eine 8-KByte-Erweiterung (oder natürlich mehr). Eine 3-KByte-Erweiterung, die man dann zusammen mit der 16-KByte-Erweiterung in eine (wohl kaum vorhandene) Modul-Box stecken muß, ist völlig unnötig. Die von mir beschriebenen POKEs reichen aus, um Programme, die eigentlich eine 3-KByte-Erweiterung benötigen, nur mit einer 8-KByte-(oder mehr)Erweiterung lauffähig zu machen.
Oliver Eichhorn

Wordpro 3+

Kann Wordpro 3+ auch mit Centronics-Druckern am Userport betrieben werden?
Ausgabe: 5/84
Heinz-Josef Erben

Die Redaktion weist in Ihrem Test in der Ausgabe 6/84, Seite 52 ff. darauf hin, daß das Programm nur mit Druckern betrieben werden kann, die am seriellen Bus angeschlossen sind.

Mir ist es nun gelungen, die nötige Treibersoftware für eine Centronicsschnittstelle in das Programm einzubinden. Darüber hinaus habe ich es auch um eine deutsche Tastaturbelegung und deutschen Zeichensatz erweitert. Wer also mit seinem Wordpro gerne »Deutsch« schreiben möchte oder einen Centronics-Drucker benutzt, kann sich gerne an mich wenden.

David Trobisch
Neckarhauserstr. 54
6800 Mannheim 71
Tel. 0621/475659

Fragen Sie doch!

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der beigehefteten Karte). Wir veranlassen, daß die Fragen von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht.

Probleme mit Monitoren

Wie schließe ich einen Monitor an den VC 20 an?
Ausgabe: 8/84
Inge Märkle

Hier mein Tip, wie ich meinen Monitor angeschlossen habe. In einem Elektrogeschäft kauft man sich einen 5poligen Diodenstecker, einen Cinch-Stecker und ein abgeschirmtes Kabel (soviel wie man eben braucht). Das alles kostet nur wenige Mark. Das Kabel wird abisoliert. Der Innenleiter wird dann an Stift 5 des Diodensteckers gelötet, das Drahtgeflecht kommt an Stift 2. Auf der anderen Seite des Kabels kommt der Innenleiter an den Stift des Cinch-Steckers, das Drahtgeflecht an den noch freien Anschluß. Das ist schon alles. Der Diodenstecker wird anstelle des Modulators in die Video-Buchse des VC 20 gesteckt, der Cinch-Stecker kommt in den Monitor. Siehe auch Seite 151 im Handbuch.

Manchmal kann es vorkommen, daß das Monitorbild trotz voll aufgedrehtem Helligkeitsregler sehr dunkel ist. Abhilfe: VC 20 aufschrauben und das Video-IC suchen. (Großes schwarzes Bauteil mit 40 Beinchen, auf dem »Mos 656« steht. Manchmal ist dieses IC von einem silbrigen Metallkästchen umgeben: Deckel entfernen.) In der Nähe dieses ICs gibt es zwei blaue Regler. Monitor anschließen und Helligkeitsregler ganz aufdrehen. Nun einen der beiden Regler vorsichtig (!) verdrehen. Verändert sich die Bildhelligkeit, so muß der Regler so lange verdreht werden, bis das Bild hell genug ist. Wenn beim Verdrehen nichts passiert, den Regler wieder in die Stellung bringen, in der er war und das Experiment mit dem anderen Regler wiederholen. Wenn auch hier nichts passiert, muß das Videosignal verstärkt werden. Den Schaltplan für einen entsprechenden Verstärker kann man bei mir bekommen.

Bernd Eßlinger
Panoramastr. 32
7470 Albstadt 2

Text und Grafik mischen

Wie kann ich in Simons Basic hochauflösende Grafik und Text gleichzeitig darstellen?
Ausgabe: 8/84
Frank Schager

Die Darstellung von Text und gleichzeitig Grafik im Grafikmodus des Simons Basic läßt sich durch den Simons Basic-Befehl »TEXT« bewerkstelligen. Dieser ist so anzuwenden:
TEXT x,y,"Text",m,gr,ab
x = X-Koordinate; y = Y-Koordinate;
m = Grafikmodus (m=0 oder m=1 oder m=2), wenn m=0, 1 oder 2) wenn m=0 dann aus;
gr = Größe des darzustellenden Buchstabens;
ab = Abstand zwischen den einzelnen Buchstaben.

Um einen gleichen Abstand und eine gleiche Größe der Buchstaben wie beim normalen Textmodus zu erreichen, wird für »gr« eine 1 und für »ab« eine 7 eingesetzt. Also:
TEXT x,y,"Text",m,1,7

Dennoch läßt sich eine Eingabe von Antworten im Grafikmodus des Simons Basic nicht sehr einfach realisieren, da bei der Eingabe, jedenfalls im Grafikmodus, weder Cursor noch Eingaben sichtbar sind. Vielleicht gibt es einen Ersatzbefehl für den herkömmlichen INPUT?
Stephan Hardy



C 64 und Videorecorder

Wie kann ich mit dem Videorecorder die Signale vom Computer aufnehmen?
Ausgabe: 8/84
Eugen Anger

Die Lösung ist ganz einfach: Man nehme das Kabel, welches beim C 64 direkt vom Computer kommt, und steckt dieses in den Eingang des Videorecorders. Dann führt man das Kabel vom Ausgang, wie in der Betriebsanleitung angegeben, zum Fernsehgerät. Nun muß man das Gerät auf den Computer einstellen, und zwar so, als würde man einen Fernsehsender einstellen. Danach braucht man nur noch wie bei der normalen Aufnahme zu verfahren.

Stefan Wößner

Hardcopy mit NEC 8023 B-N

Wer bietet Software an, um mit dem C 64 und dem Drucker NEC 8023 B-N hochauflösende Grafik und Hardcopys zu erstellen?
Ausgabe: 8/84
Wolfgang Jaworski

Unser neues Interface Typ 92000/NEC erlaubt, den NEC-Drucker 8023 oder Itoh 8510 an den C 64 anzuschließen. Der Drucker wird voll kompatibel zum CBM-Drucker 1515. Auf diese Weise können zum Beispiel mit Simons Basic problemlos Hardcopys erstellt werden. Das Interface ist für 298 Mark inklusive Mehrwertsteuer sofort lieferbar. Vorhandene Geräte Typ 9200 rüsten wir für 91,20 Mark inklusive Mehrwertsteuer in die neue Version um. Ihr Leser

Reinhard Wiesemann
Mikrocomputertechnik,
Wuppertal

Rechengenauigkeit beim C 64

Warum gibt der C 64 als Ergebnis der Aufgabe PRINT INT(3/0.03) nicht 100, sondern 99 an?
Ausgabe: 9/84
Oliver Treiber

Wie genau der C 64 in seiner Ausarbeitung ist, beweist Ihr Auftrag-Print INT(3/0.03). Von der Logik her ist diese Antwort als richtig einzustufen. Die INT-Funktion (Ganzzahl-Funktion) geht davon aus, daß im Endergebnis die kleinere, ganze Zahl ausgegeben wird. Das Ergebnis lautet natürlich 100; da aber die INT-Funktion die nächst kleinere Zahl anzugeben hat, wäre dies die Zahl 99.99, die der 100 am nächsten käme.

Nun hat aber die INT-Funktion auch die Aufgabe, aus einer Zahl mit Dezimalstellen eine ganze Zahl zu machen. Also werden die beiden Dezimalstellen geschluckt. Das INT-Ergebnis lautet daher 99. Nun kann man diese »Ungenauigkeit« in der INT-Funktion umgehen, in dem man das Originalergebnis mit einer Dezimalstelle versieht. PRINT INT(3/0.03+0.5). Das Ergebnis müßte jetzt 100.5 lauten. Aber die nächst kleinere, ganze Zahl lautet 100.

Im übrigen kann man mit der INT-Funktion auf- und abrunden sowie vorausbestimmen, auf wieviele Dezimalstellen das Endergebnis ausgedruckt werden soll. Im Prinzip kommt es darauf an, welche Aufgabe eine Funktion übernimmt. Eine solche Rechenoperation löst man einfacher mit Print 3/0.03. Besondere Vorsicht ist bei einer INT-Funktion mit einer negativen Zahl geboten.

Prüfen Sie die Aussage PRINT INT(—2.3) auf die Aussage der nächst kleineren, ganzen Zahl hin. Das Ergebnis lautet —3.

Rolf Voigt

Programmunterbrechung bei Druckerausgabe

Bei Ausgabe über Drucker erfolgt nach unterschiedlich langer Druckerausgabe des öfteren die Fehlermeldung »Device not present error«. Handelt es sich um einen Hard- oder Softwarefehler?
Ausgabe 8/84
Rudolf Ott

Als Hersteller von Zusatzgeräten für den C 64 ist uns das geschilderte Problem bekannt. Der Fehler liegt im seriellen Bus des C 64. Er dürfte nicht mehr auftreten, wenn die Signalleitungen an der 6poligen Buchse mit 4,7 kOhm gegen Masse abgeschlossen sind.

Josef Wildgruber
Fotronic GmbH

Autostart

Wie bringe ich meine Programme dazu, daß sie nach dem Laden von Diskette oder Kassette automatisch starten?
Ausgabe: 6/84
Carsten Bruch

Ich habe noch eine ergänzende Antwort. Um einen Autostart bei der Floppy hervorzurufen, genügt es, folgende Zeile einzugeben:
LOAD"NAME",8:(SHIFT)(RUN/STOP)

Dabei müssen die Tasten (SHIFT) und (RUN/STOP) gleichzeitig gedrückt werden. Das Programm »NAME« wird dann ohne Eingabe von (RETURN) automatisch geladen und gestartet.

Arndt Grothoff

Bits hörbar machen

Ist es beim C 64 möglich, beim Laden von Datasette die Signale hörbar zu machen?
Ausgabe: 8/84
Werner Frings

Um die Signale, die von der Datasette kommen, hörbar zu machen, muß einfach mit POKE54296,15 die Lautstärke im SID auf 15 gesetzt werden. Nun sind die Signale beim Laden, Saven und Verifyen über den Lautsprecher des Monitors zu hören (auch bei Fast- und Turbo Tape).

Und nun ein Nachtrag zum Autostart: Es gibt auch eine einfachere Lösung als die von Ihnen beschriebene.

1. Programm einladen (Disk oder Tape)
2. PRINT PEEK(174),PEEK(175)
3. Beide Werte notieren
4. Als erste Anweisung im Programm POKE174,<1.Wert>:POKE175,<2.Wert> setzen
5. Programm abspeichern
6. NEW
7. Ein kleines Ladeprogramm entwerfen:
10 LOAD''Name>',8 (bei Disk oder
10 LOAD''Name>' (bei Kassette)
8. Dieses Miniprogramm abspeichern (bei Kassette vor dem Hauptprogramm)

Soll nun das Hauptprogramm mit Autostart geladen werden, muß das Ladeprogramm geladen und mit RUN gestartet werden. Der Computer sucht darauf das Hauptprogramm, lädt und startet es.

Nun zur Funktionsweise: Wenn in einem Programm eine LOAD-Anweisung erfolgt, lädt der Computer schon das angesprochene Programm und startet es. Probleme tauchen allerdings auf, wenn das zu ladende Programm länger ist als das Ladeprogramm. Der Computer lädt zwar die Programmdaten mit, »glaubt« sich aber immer noch im Ladeprogramm, da die Zeiger aufs Programmende sich in diesem Fall nicht dem neuen Programm anpassen. Die Adressen 174 und 175 enthalten die Zeiger aufs Programmende und können so mit den POKEs in der ersten Zeile angepaßt werden.

Dieses Verfahren läuft sowohl auf Diskette als auch auf Kassette, allerdings nicht mit Fast- oder Turbo Tape.

Stefan Gossens

EPROM-Brenner: Disput

Mit Interesse las ich Ihren Testbericht über EPROM-Programmer in der 64'er, Ausgabe 8/84.

Doch schon bald wurde aus dem Interesse Unmut. Unmut über einen zu oberflächlichen Test, der wenig die Funktion der Geräte ausgeleuchtet hat.

Als Hersteller der Geräte Mod. V64 und Mod. V128 (Fa. Jeschke tritt als Wiederverkäufer dieser Geräte auf), möchte ich hier ein wenig das Konzept erläutern: Die EPROM-Programmer sind für den Einsatz am User-Port vorgesehen. Das hat drei große Vorteile:

1. Ein Gerät läuft ohne Änderung an der Hardware auf zwei Computern (nämlich dem VC 20 und dem C 64)
2. bleibt der Expansionport frei für Steckmodule, 80-Zeichenkarten etc.
3. Ist es am User-Port möglich, über die herausgeführte Wechselspannung auf einfache Art und Weise die Programmierspannung zu erzeugen. Ein externes Netzteil ist somit nicht nötig, was zumindest ein lästiges Kabel erspart, letzten Endes aber für den Kunden Sicherheit darstellt, da er nicht mit der Netzspannung in Berührung kommen kann.

Da vor allem die Software unserer Geräte bemängelt wurde, auch hierzu eine Stellungnahme:

Es ist nicht sinnvoll, für die Treibersoftware Maßstäbe an-

zusetzen wie etwa für ein Telespiel.

Grafik, Farbe und Ton können zwar ganz reizvoll sein, sind aber für die Funktion der Software nicht notwendig und bei guter Dokumentation sogar überflüssig.

Auch die Hintergründe für bestimmte Befehle der Treibersoftware wurden im Test vernachlässigt:

Eine Transfer- (Verschiebe-) Funktion ist in unserer Software nicht eingebaut, da sie nicht notwendig ist. Anders als bei anderen Programmiergeräten können Sie den Speicherbereich, aus dem das Quellprogramm kopiert werden soll, frei wählen. Dabei ist es auch möglich, die Programmlänge beliebig festzusetzen, sowie die Startadresse im EPROM frei zu wählen. Unsere Treibersoftware gestattet Ihnen so das Programmieren eines »halben« EPROMs, oder sogar nur ein einzelnes Byte zu programmieren. Das Nicht-Vorhandensein von LOAD- und SAVE-Routinen hat folgende Gründe:
Aufgabe der Treibersoftware ist in erster Linie die Bedienung des Programmiergerätes. So ist zum Beispiel das Kopieren eines EPROMs mit der Treibersoftware kein Problem.

Für das umfassende Arbeiten mit Maschinenprogrammen benötigt der Anwender aber ein Monitorprogramm. Da ein Monitorprogramm normalerweise LOAD- und SAVE-Routinen enthält, brauchen sie in der Treibersoftware kein zweites Mal installiert zu sein.

Für solche Fälle kann ich nur unsere UNIMENT-Befehlserweiterung empfehlen, die neben zahlreichen neuen Basic-, Grafik-, Sprite- und Sound-Befehlen sowie einer parallelen Centronics-Schnittstelle einen kompletten Maschinenmonitor mit Assembler-, Lade-, Speicher-, Transfer-Routinen und so weiter enthält.

Auch die Treibersoftware für unsere EPROM-Programmer ist bereits implementiert, so daß beim UNIMENT-Steckmodul keinerlei Software geladen werden muß. Mit dem Einschalten des Computers stehen dem Benutzer die oben genannten Funktionen sofort zur Verfügung.

Hagen Völzke

Als Autor des EPROM-Vergleichstests in der Ausgabe 8/84 kann der oben abgedruckte Brief der Firma Völzke meinerseits nicht unbeantwortet bleiben.

Der erste Unmut stellte sich bei Herrn Völzke wegen eines angeblich zu oberflächlichen Tests ein. Dazu ist folgendes zu sagen:

1) Ich besitze persönlich seit über einem halben Jahr das Modell V128 und benutze es regelmäßig. Von einer Unkenntnis des Gerätes kann also sicher nicht die Rede sein.
2) Die meisten von H. Völzke bemängelten Punkte werden im Test angesprochen. Dazu im einzelnen: Die Anschlußmöglichkeit der EPROM-Brenner am User-Port ist aus der Vergleichstabelle auf Seite 37 Spalte 2 ersichtlich. Dort kann ebenfalls gelesen werden, daß die EPROMmer der Firma Völzke sowohl für den VC 20 als auch den C 64 eingesetzt werden können. Als wesentliches Entscheidungskriterium für die Testbewertung ist die Tauglichkeit für beide Computer aber sekundär, da nur sehr wenige Computeranwender einen VC 20 und einen C 64 besitzen. Die Anzahl derer, die dann noch EPROMs brennen, scheint mir verschwindend gering.

Der zweite Kritikpunkt bezieht sich auf das Freibleiben des Expansion-Ports bei den Völzke-Geräten. Nun, daß es auch bei einem EPROM-Brenner mit Anschluß am Expansion-Port zu keinem Verlust des Expansion-Ports kommen muß, zeigt das Gerät von Kalawski.

Der beste Beweis dafür, daß auch ich den Anschluß einer zusätzlichen Stromversorgung nicht für notwendig halte, ist in der Tatsache zu sehen, daß der Testsieger von Roßmöller ebenfalls ein User-Port Gerät ist.

Ein paar Worte sollten aber auch zur Software hinzugefügt werden:

1) Natürlich ist die Treibersoftware für ein EPROM-Programmiergerät nicht mit einem Telespiel zu vergleichen. Dennoch betrachte ich die optische und akustische Hilfen geführt für den Benutzer als sinnvoll. Bedenkt man den doch relativ hohen Preis für EPROMs, so ist die Zerstörung durch Fehlprogrammierung oder falsches Einsetzen für sicherlich fast jeden Anwender ein herber Verlust. Auch ist das wiederholte Löschen und Neuprogrammieren eines falsch »geschossenen« EPROMs mit dem Nachteil der verkürzten Lebensdauer des EPROMs verbunden.

Die Treibersoftware der Völzke-Geräte bietet in diesem Bereich keinerlei Sicherheit (ein Freund hat wegen einer falsch eingestellten Programmierspannung einen Verlust von über 600 Mark gehabt). Viel besser in Hinsicht auf die Sicherheit ist der Testsieger und das Gerät von Kalawski, wobei letzterer sogar den richtigen Einsetzpunkt des EPROMs optisch anzeigt.
2) Auch andere EPROM-Brenner verfügen über die Fähigkeit, den Speicherbereich, aus dem das Quellprogramm kopiert werden soll, frei zu wählen.
3. Die weiterhin von Herrn Völzke angesprochenen Punkte, wie die LOAD- und SAVE-Funktion oder der Einsatz eines Monitors für diese Zwecke sind zwar richtig, er vergißt aber, daß für ein solches Programm zusätzlich finanzielle Aufwendungen entstehen. Viel besser ist es doch, wenn diese Funktionen, wie bei unserem Testsieger, kostenlos mitgeliefert werden, da sie in der Treibersoftware implementiert sind.
4) Herr Völzke empfiehlt für oben genannte Zwecke den Einsatz seiner Befehlserweiterung UNIMENT. Dafür sind allerdings nochmals 99 Mark auf den Tisch zu legen: Der Vorteil dieses Konzeptes soll aber nicht verschwiegen werden. Sowohl die UNIMENT-Erweiterung als auch der Monitor (der aus einem riesigen Softwareangebot ausgewählt werden kann) sind auch ohne den EPROMmer einsetzbar. Rechnet man zum Preis der Völzke-Geräte den eines guten Monitors hinzu, so werden zirka 300 Mark fällig. Dafür bekommt man aber auch schon unseren Testsieger mit eingebautem Monitor und einem überlegenen Konzept, das beispielsweise die Programmierung der zukunftsträchtigen 27256-EPROMs ermöglicht.

(Arnd Wängler)

Vielseitiges Supervoc Ausgabe: 6/84

Alle Besitzer einer VC 1541, die Vokabeln mit dem Programm Supervoc lernen wollen, können sich glücklich schätzen. Das Programm muß, um auch mehrere Vokabelprogramme zu laden oder gleichzeitig zu schreiben, folgendermaßen geändert werden:
In Zeile 220 Diskette geschrieben, statt Kassette. In Zeile 230 muß der erste GOSUB-Befehl in GOSUB 1315 umgeändert werden.

```
1315 PRINT »Welches Programm
soll geladen werden?«
1317 INPUT P$
1330 OPEN 1,8,0,P$
1340 INPUT #1,P$
1470 OPEN 1,8,1,N$
```

Außerdem: Wer kann mir einen Tip geben, wie man das Kopierprogramm (Disk Copy), aus der April-Ausgabe, zum Laufen bekommt? Gibt es ein Vokabel-Lernprogramm für französische Vokabeln?

Markus Bluhm
Gruberzeile 95
1000 Berlin 20

Autostart geknackt Ausgabe 8/84, Seite 98

Dirk Rother irrt sich, wenn er behauptet, Autostartprogramme könnte man nicht mehr stoppen und überarbeiten. Ein Autostartprogramm kann wie folgt geknackt werden:

Kassette in normalen Recorder einlegen und Autostartprogramm anhören, Kassette in Datasette einlegen, Counter auf Null stellen, Kassette auf Anfang des Autostartprogrammes zurückspulen, Autostartprogramm einladen, wenn Counter auf Null steht dann Stopp-Taste an Datasette drücken, andere Kassette mit dem vorher abgeSAVEten Kurzprogramm: 10PRINT" ": END einlegen und Play-Taste drücken. Nach ein paar Sekunden erscheint auf dem Bildschirm READY. Das Autostartprogramm läßt sich listen, das nachfolgende Programm einladen, überarbeiten und kopieren. Das Autostartprogramm kann auch wieder abgespeichert werden.

Ich glaube mit an Sicherheit grenzender Wahrscheinlichkeit, daß Ihr es Euch nicht traut, den Leserbrief abzudrucken. Aber warum denn nicht? Ihr bringt so viele Listings wie man das und jenes kopieren kann und ein paar Seiten weiter:
»Wir wollen unsere Leser darauf aufmerksam machen, daß verkaufen, tauschen, etc.«. Das ist so wie: Hier hast du den Schlüssel zum Tresor, aber aufschließen darfst du ihn nicht. Überlegt doch mal selber oder versucht es zumindest einmal!

Ein Knackie

Lieber Knackie, Sie sehen, wir haben nachgedacht und uns getraut.

Rettung, wenn ISM 64 abstürzt

Bei der Anwendung des Dateiverwalters ISM ist es mir bei einer damit verbundenen Druckerbenutzung auch passiert, daß der ISM abstürzte. Nach einigem Herumprobieren bin ich auch darauf gekommen, daß durch ein Initialisieren der Diskette der Betrieb fortgesetzt werden kann. Diese Lösung hat allerdings in keiner Weise befriedigt, und so wandte ich mich an den Hersteller, die SM-Software AG. Nach einer Woche bekam ich die erlösende und sehr einfache Antwort.

Nachdem der Druckerkanal geöffnet wurde, muß vor dem nächsten ISM-Befehl der Basic-Befehl POKE 186,GN eingegeben werden. GN ist dabei die Gerätenummer der Floppy, also meistens 8.

Frank Götze

MEHR ÜBERSICHT

Mit 506 Zeichen auf dem Bildschirm sind die Grenzen recht eng gesetzt. VC 20-Besitzer konnten nur neidisch auf den C 64 mit 1000 Zeichen oder den CBM 8032 mit 2000 Zeichen schauen. Mit einer 40/80-Zeichenkarte kann es der VC 20 jedoch leicht auch mit diesen Computern aufnehmen.

Um 40 oder 80 Zeichen pro Zeile mit dem VC 20 zu realisieren, ist auch eine Software-Lösung möglich. Sie ist jedoch sehr aufwendig, da neben dem Zeichensatz auch weite Teile des Betriebssystems geändert werden müssen. Außerdem ist diese Lösung sehr langsam und nicht für professionelle Anwendungen geeignet.

Eine Hardware-Lösung ist dagegen leichter zu handhaben (nur Karte einstecken) und bietet zusätzliche Leistungen. Wir haben die 40/80-Zeichenkarte MR 40/80 (Bild 1) von Roos-Elektronik getestet.

Wie bedient man nun diese 40/80-Zeichenkarte, und was kann sie?

Aus der sehr mageren Beschreibung kann man leider nur wenig entnehmen. Die MR 40/80 stellt jedenfalls keine Farben mehr am Bildschirm dar. Reverse Schrift ist jedoch weiterhin möglich. Im 40-Zeichenmodus sind sowohl die Schrift wie auch die Grafikzeichen am Fernseher noch einwandfrei lesbar (Bild 2). Bei 80 Zeichen je Zeile tut man sich jedoch am Fernseher recht schwer (Bild 3), deshalb ist ein Monitor unbedingt zu empfehlen.

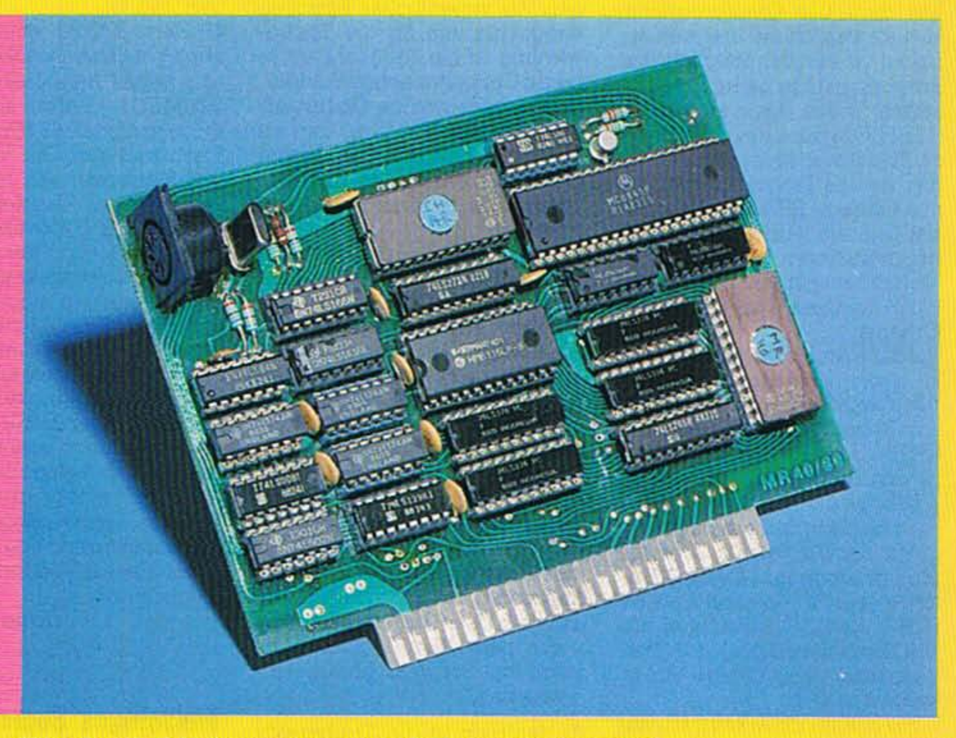


Bild 1. Die MR 40/80-Karte macht aus dem VC 20 ein professionelles Gerät

Nach dem Einstecken der MR 40/80 in den Expansionport des VC 20 wird an der fünfpoligen DIN-Buchse der Karte ein Monitor oder der HF-Modulator für den Fernseher angeschlossen. Nach dem Einschalten meldet sich der VC 20 wie gewohnt, jedoch mit einer Bildschirmbreite von 40 Zeichen und ohne Farbe. Das Bildschirmformat kann durch Drücken bestimmter Tasten während des Einschaltens geändert werden: Eine der Tasten 0 bis 8 bestimmt die Anzahl der Punkte für den Zeilenabstand. Diese Funktion ist ähnlich dem CHR$(14) beim CBM 4001/8001, das den Zeilenabstand um zwei Punkte vergrößert. Betätigt man eine der Tasten 0 bis 8 in Verbindung mit der rechten SHIFT-Taste, dann wird der 80-Zeichenmodus aufgerufen.

Drücken der RETURN-Taste während des Einschaltens verhindert, daß die MR 40/80 aktiviert wird. Die Bildschirmausgabe erfolgt dann über den bisherigen Video-Ausgang des VC 20, also bitte das Umstecken des Kabels nicht vergessen.

Um das Bildschirmformat bei eingeschaltetem Computer zu ändern, drückt man eine der oben beschriebenen Tastenfolgen zusammen mit der RESTORE-Taste. Das Umschalten in den 80-Zeichenmodus ist aber etwas kompliziert: Man muß gleichzeitig die rechte SHIFT-Taste, die CTRL- und die RESTORE-Taste drücken.



Bild 2. Die 40-Zeichen-Darstellung ist am Fernsehgerät oder an einem einfachen Monitor gut erkennbar.

AM BILDSCHIRM

Test 40/80-Zeichenkarte

Zurück in den 40-Zeichenmodus kommt man durch Betätigen von CTRL und RESTORE. Das Programm bleibt beim Hin- und Herschalten natürlich erhalten.

Sonderfunktionen auf den Funktionstasten

Alle acht Funktionstasten werden durch die MR 40/80 mit speziellen Sonderfunktionen für die Textverarbeitung belegt (Tabelle 1). Unter anderem sind ein Tabulator und zusätzliche Lösch- und Cursorsteuerfunktionen vorhanden. Alle diese Funktionen F1 bis F8 können natürlich auch vom Programm aus aufgerufen werden. Dies geschieht mit »PRINT CHR$(133 ... 140)«.

Zu beachten ist die Lage des Bildschirmspeichers. Die MR 40/80 enthält ein eigenes Video-RAM von 2 KByte Umfang, das ab Speicheradresse 43008 ($A800) liegt. Da der normale Video-Speicher nicht benötigt wird, stehen zusätzliche 512 Bytes RAM zur Verfügung. Mit SYS 43000 wird dieser Speicher dem Basic-RAM hinzugefügt.

Für viele Programme ist es durchaus sinnvoll, die oberste Zeile des Bildschirms zu schützen. Die Zeile bleibt dann unverändert stehen und scrollt auch nicht aus dem Bildschirm heraus. So bleiben Überschriften oder Bedienungshilfen ständig sichtbar.

Die MR 40/80-Karte ist recht übersichtlich aufgebaut und mit einem Video-Controller, einem 2 KByte-CMOS-RAM sowie mit zwei EPROMs (2716 und 2732) und diversen Logikbausteinen bestückt. Alle IC sind gesockelt (!) und können im Schadensfall leicht selbst ausgewechselt werden. Die Steuersoftware im 2716-EPROM belegt den Adreßbereich von $A000 bis $A7FF. Im 2732-EPROM befindet sich der

Aufbau der Platine

Zeichengenerator, der natürlich auch leicht ausgewechselt werden kann. Für die Textverarbeitung wären beispielsweise deutsche Umlaute recht sinnvoll. Ein entsprechendes EPROM ist auf Anfrage beim Hersteller erhältlich.

Die Platine macht einen recht stabilen Eindruck. Leider hat es für ein Gehäuse bei dem immerhin 249 Mark teuren Gerät nicht mehr gereicht.

Fazit: Diese 40/80-Zeichenkarte ist für professionelle Anwendungen durchaus geeignet. Die 40 Zeichen pro Zeile sind am Fernseher gut lesbar, die 80 Zeichen Darstellung jedoch ist ohne Augenschäden auf Dauer nur mit einem Monitor möglich. Da die Farbe fehlt, ist die Karte für alle Arten von Spielen weniger geeignet.

(Christian Q. Spitzner/ev)

```
10 PRINT"NACH DEM EINSCHALTEN DES VC 20 MELDET ER SICH WIE GEWOHNT, JEDOCH MIT 4
0";
20 PRINT" ZEICHEN BILDSCHIRMBREITE UND OHNE FARBE. NOCH    WAEHREND DER";
30 PRINT" EINSCHALTPHASE KANN MAN    DURCH BESTIMMTE TASTENKOMBINATIONEN DAS";
40 PRINT" BILDSCHIRMFORMAT BESTIMMEN:
45 PRINT
50 PRINT"EINE DER TASTEN 0 ...8 BESTIMMT DIE    ANZAHL DER DOTS FUER DEN";
60 PRINT" ZEILENABSTAND  (9 BEDEUTET KEIN DOT- LINE- SPACING).
70 PRINT
80 PRINT"DIESE FUNKTION IST AEHNLICH DES CHR$ (14) BEIM CBM 4001 BZW 8001,  ";
90 PRINT" WODURCH DER ZEILENABSTAND UM ZWEI DOTS VERGROESSERT WIRD."
100 GOTO100
READY.
RUN
NACH DEM EINSCHALTEN DES VC 20 MELDET ER SICH WIE GEWOHNT, JEDOCH MIT 40 ZEICHEN
 BILDSCHIRMBREITE UND OHNE FARBE. NOCH    WAEHREND DER EINSCHALTPHASE KANN MAN
 DURCH BESTIMMTE TASTENKOMBINATIONEN DAS BILDSCHIRMFORMAT BESTIMMEN:

EINE DER TASTEN 0 ...8 BESTIMMT DIE    ANZAHL DER DOTS FUER DEN ZEILENABSTAND
(9 BEDEUTET KEIN DOT- LINE- SPACING).

DIESE FUNKTION IST AEHNLICH DES CHR$ (14) BEIM CBM 4001 BZW 8001,   WODURCH DER
ZEILENABSTAND UM ZWEI DOTS VERGROESSERT WIRD.
```

Bild 3. Bei 80 Zeichen pro Zeile wird es bei einem Fernseher kritisch. Hier hilft nur ein monochromer Monitor der höheren Preisklasse.

Tabelle 1. Die Belegung der Funktionstasten

F1 setzt den Cursor auf die letzte Bildschirmzeile
F2 schaltet die MR 40/80-Karte ohne Programmverlust aus
F3 löschte den Bildschirm hinter dem Cursor
F4 bewirkt das Einfügen einer Zeile an der Cursorposition
F5 springt zur nächsten Tabulatormarke
F6 erweitert das Basic-RAM um die 512 Bytes des Video-RAM
F7 löscht alles rechts vom Cursor
F8 löscht alle Zeichen links vom Cursor

Gesamtüberblick	
Geschwindigkeit:	sehr gut
Bedienung .....:	sehr gut
Beschreibung ...:	mäßig
Bildqualität (40) :	sehr gut
Bildqualität (80) :	gut
Gesamturteil ....:	gut



X 100 — Farbig plotten und drucken

Seitdem moderne Matrixdrucker voll grafikfähig sind, haben Plotter es schwer, sich durchzusetzen. Auch der relativ hohe Preis wirkte leicht abschreckend. Der X 100 ist jedoch von der Leistung als auch vom Preis, für den C 64-Benutzer interessant.

Einen Plotter in Aktion zu sehen, ist ein aufregendes Erlebnis. Das beweisen die Menschentrauben, die zum Beispiel bei Messen und Fachausstellungen so einem Gerät bei der Arbeit fasziniert zuschauen. Man überlegt sich unwillkürlich, ob man sich nicht auch so einen Zeichenkünstler anschaffen sollte. Doch wenn dann Preise genannt werden, sinkt die Kauflust oft ganz erheblich. Aber ebenso wie die Matrixdrucker, brechen auch die Plotter in einen bisher ungesättigten Markt ein: Sie werden immer preiswerter, bieten aber trotzdem beachtliche Leistungen. Der Adcomp X100 ist solch ein Gerät (Preis: zirka 2000 Mark).

Auf den ersten Blick verwechselt man ihn glatt mit einem üblichen Matrixdrucker. Ein ähnliches Gehäuse, ein Handrad an der Seite, Tractoreinheit und Gummiwalze. Doch schon ein etwas genauerer Blick unter die Gehäuseabdeckung zeigt den wichtigen Unterschied auf: Anstelle des Druckkopfes sieht man 4 Farbstifte, die in einem Revolvereinsatz stecken. Spätestens jetzt wird jedem klar, daß er einen Plotter vor sich stehen hat.

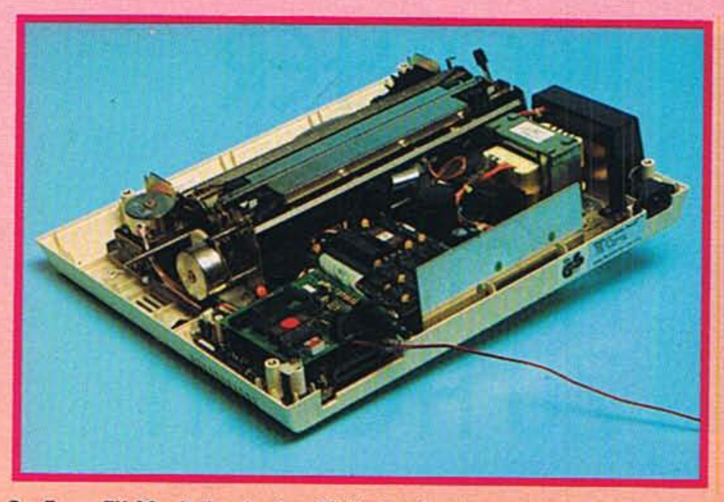
Der X100 wird mit einer deutschen Anleitung geliefert. Speziell für den C 64-Besitzer liegt noch eine zusätzliche Broschüre bei, die auf die speziellen Befehle des Plotters und auf die Programme der ebenfalls beiliegenden Demodiskette eingeht und sie beschreibt. Außerdem befindet sich auf der Diskette ein Druckertreiber-Programm, das die Centronics-Schnittstelle des Plotters ansteuert. Der Plotter wird mit einem Kabel über den User-Port des C 64 verbunden. Die Treibersoftware entspricht übrigens dem Programm aus Heft 7.

Aktiviert wird der X100 über die bekannten OPEN n,4-Befehle wie bei den Matrixdruckern. Damit ist es auch möglich, Texte zu plotten, also auch Listings. Allerdings können keine Commodore-Grafik- und -Steuerzeichen dargestellt werden. Kommen wir jedoch zu den Plottmöglichkeiten.

Es können selbstverständlich Linien in jeder Richtung gezeichnet werden. Dabei ist sowohl eine relative als eine absolute Bewegung möglich. Linien lassen sich durchgezogen, aber auch gestrichelt zeichnen. Der Abstand zwischen den Teilstrichen kann eingestellt werden. Text läßt sich an jeder beliebigen Stelle setzen. Auch die Schriftgröße und die Schreibrichtung sind definierbar, wobei jedes Zeichen zwischen 1 und 255 mm groß sein und in 4 Richtungen gedreht werden kann. Oft benötigte Routinen wie Achsen, Kreise und Histogramme (Rechtecke) sind auch implementiert; umständliche Unterprogramme können deswegen entfallen. Die Lage und Parameter der einzelnen Figuren sind frei wählbar. Selbstverständlich kann man zu jeder Zeit zu einer der anderen vier Farben wechseln. Die Farbstifte, mit Druckflüssigkeit gefüllte Kugelminen, sind ausreichend dünn. Sie besitzen laut Handbuch eine »Lebensdauer« von mehr als 800 Meter. Auch in der kleinsten Schriftart (1 mm) sind die Zeichen noch gut zu lesen (mit Lupe). Das bewirkt aber auch die Schrittweite von wahlweise 0,1 beziehungsweise 0,05 mm. Die Wiederholgenauigkeit beträgt 0,2 mm bei Positionierung über 50 mm und die Positionierungsgenauigkeit erreicht zirka 1% bei einer Positionierung über 100 mm. Die Plottgeschwindigkeit reicht von 120 mm/sec, bei 0,1 mm Schrittweite, bis 60 mm/sec bei 0,05 mm Schrittweite. Der effektive Plottbereich liegt in beiden Hauptrichtungen (X- und Y-Achse) bei 20 cm.

Bewertung

Der X100 ist ein Plotter nicht nur für den Hobby-Einsatz. Durch sein Papierformat, DIN A4, mit wahlweiser Verwendung von Einzelblatt oder Endlospapier wird er vielseitig einsetzbar. Seine technischen Daten genügen gehobenen Ansprüchen, auch wenn sie selbstverständlich nicht an die echte Profiklasse heranreichen. Ein kleiner Tip noch für VC 1520-Besitzer: Alle Befehle des VC 1520 werden auch vom X 100 verstanden. Lediglich einige Längenangaben müssen wegen dem unterschiedlichen Plottbereich etwas geändert werden. (gk)



Der Epson FX 80 mit eingebautem Görlitz-Interface

Ein Drucker für alle Fälle: Epson FX 80

Der FX 80 ist ein Matrixdrucker, der kaum noch Wünsche offen läßt. Zur Zeit seiner Einführung eine Sensation, gehört er aber auch heute noch immer zu den Besten seiner Klasse.

Jeder C 64-Besitzer wird sich irgendwann einmal einen Drucker wünschen, entweder, um seine selbst erstellten Programme auszudrucken, oder um Briefe, Listen und andere wichtige Dinge zu Papier zu bringen. Es stellt sich dann die Frage, was der Drucker leisten soll. Nicht zuletzt hat da das eigene Konto einen wichtigen Diskussionsbeitrag zu leisten.

Eines kann schon vorausgeschickt werden: Es wird schwer fallen, alle Möglichkeiten des FX 80 auszunutzen. Seine vielen Druckfunktionen dürften auch die meisten Textverarbeitungsprogramme überfordern. In der Praxis wird es jedoch so sein, daß man sich einige wichtige Schriftarten merkt und nur mit diesen arbeitet. So kann man mit einfachen Befehlen zum Beispiel Breitschrift, Schmalschrift, Elite, Fettdruck, Doppeldruck, Proportionaldruck und Kursivschrift wählen. Viele dieser Druckarten sind auch untereinander mischbar. Natürlich läßt sich auch der Zeilenabstand fast stufenlos verstellen und der Papiertransport durch einfache Steuersignale variieren. Tabulatoren lassen sich vertikal und horizontal einstellen, ebenso wie der linke und rechte Druckrand. Wem der Drucker zu schnell ist, der kann ihn durch einen Befehl auf halbe Geschwindigkeit drosseln. Die neun internationalen Zeichensätze ASCII, französisch, deutsch, englisch, amerikanisch, schwedisch, dänisch, japanisch und italienisch lassen sich ebenso einstellen wie die Papierlänge, die in Zeilen oder Zoll angegeben werden kann. Wem die internationalen Zeichensätze nicht reichen, der kann sich selbst welche entwickeln. Der frei ladbare Zeichengenerator ist 2 KByte groß und kann auch abgeschaltet werden. Falls kein eigener Zeichensatz geladen wird, dient der Speicher als Druckerpuffer. Dadurch kann man schon mit dem Computer weiterarbeiten, wenn der FX noch zu tun hat. Die Grafikfähigkeit des Epson braucht eigentlich nicht weiter betont zu werden. 64'er-Leser finden ab und zu einige Beispiele einer hochauflösenden Grafik, die der FX 80 gedruckt hat.

Innerhalb seines Gehäuses befinden sich zwei DIP-Schalter, mit denen sich einige Möglichkeiten fest einstellen lassen, so daß nach dem Einschalten der gewünschte Zustand anliegt. Folgende Schalter können gesetzt werden: Einstellung der Zahl der Druckspalten (80 und 132 Spalten pro Zeile), Form der Null (durchstrichen oder nicht), Papierendeerkennung an und aus, Sperren oder freigeben des Zeichengenerators, Fettdruck oder Normaldruck nach dem Einschalten sowie die Wahl des Internationalen Zeichensatzes (1 bis 9). Der zweite DIP-Schalter schaltet den Summer, den automatischen Seitenvorschub und den automatischen Zeilenvorschub an und aus. Außerdem kann ein Schalter gesetzt werden, der es erlaubt, den Drucker per Software anzusteuern oder abzuschalten. Leider sind diese Schalter nur nach dem Öffnen eines Gehäusedeckels erreichbar, jedoch hat man damit keine Schwierigkeiten.

Etwas, was uns sehr gefällt, aber auch manchmal die Tränen in die Augen schießen läßt, ist die Handhabung und Bedienung. Was uns sehr gefällt, ist die Papierabrißkante. Sie erweist sich in der Praxis als sehr vorteilhaft, weil lediglich zirka 2 cm über dem Druckkopf das Papier an einer Metallschiene abgerissen werden kann. Kein lästiges Vorwärtstransportieren um eine halbe Seite und damit Blattverlust und anschließendes Neujustieren der Perforationskante. Auch Zurückdrehen des Papiers ist ohne Probleme möglich. Nur das Einziehen von neuem gelochtem Endlospapier über den Tractor erfordert manchmal eine Engelsgeduld. Wenn es nicht auf Anhieb klappt und sich eventuell sogar Papierfetzen zwischen den schwer zugänglichen Tractoren verfangen, geht die Fummelei mit Pinzette oder anderen spitzen und dünnen Werkzeugen los. Um solchen Ärger nicht erst aufkommen zu lassen, muß man beim Papierabzug sehr vorsichtig und akkurat vorgehen. Dann klappt es allerdings sehr gut. Beim Einzelblatteinzug hingegen gibt es gar keine Probleme.

Der FX 80 ist ein Drucker, der es in sich hat. Bezogen auf seinen Preis (um 1500 Mark ohne Interface) und seine Nutzungsfähigkeiten ist er ein Matrixdrucker, der keine Konkurrenz zu scheuen braucht. Durch seine Verbreitung haben sich auch schon mehrere Fremdhersteller bemüht, entsprechende Schnittstellen zum C 64 zu entwickeln. Mit dem richtigen Interface können nicht nur die Fähigkeiten des FX voll ausgenutzt, sondern sogar noch erweitert werden.

(gk)



BROTHER HR-5C: FAST NICHT ZU HÖREN

Der Brother HR-5C ist ein Thermodrucker, der aber auch mit normalem Papier arbeiten kann. Er ist zwar keine Schönheit, aber dafür klein und vor allem sehr leise.

Mit den Maßen 303(B)*174(H)*65 (T) mm und einem Gewicht von 1,6 kg gehört er zu den ganz kleinen. Er paßt somit in die kleinste Ecke. Auch sonst wird der HR-5C keine großen Schwierigkeiten bereiten. Das fängt bei der Stromversorgung an: wahlweise über ein Netzteil oder über Batterien, die mitgeliefert werden. Falls diese durch häufige Benutzung verbraucht sein sollten, ertönt ein eingebauter Summer, der zum Batteriewechsel auffordert. Im Gegensatz zu den anderen »Brüdern« besitzt der HR-5C jedoch keine Tastatur. Damit ist die Wahlmöglichkeit Batterie oder Netzteil nicht mehr so interessant. Denn der Computer muß ja angeschlossen sein, um den Drucker arbeiten zu lassen. Und dieser Anschluß ist eine große Stärke der Brother Drucker. Sie besitzen eine Schnittstelle, die voll kompatibel ist mit dem Commodore-Zeichensatz. Das schließt sämtliche Grafikzeichen ein. Sie werden so gedruckt, wie sie auch auf dem Bildschirm zu sehen sind.

Da der HR-5C ein Thermodrucker ist, wird wärmeempfindliches Papier benötigt. Es sei denn, man legt eine mitgelieferte Farbbandkassette ein. Dann ist es auch möglich, auf normales Papier zu drucken. Allerdings sollte darauf geachtet werden, möglichst glattes Papier zu verwenden. Papier mit rauher Oberfläche (wie zum Beispiel teures Dekorationspapier mit Textilfasern) kann zu Problemen führen. Es kann Papier auf Rollen verwendet werden (eine volle Rolle wird mitgeliefert), aber auch Einzelblatteinzug ist möglich. Und der Papiereinzug ist denkbar einfach und problemlos. Auch wird ein akustisches Signal gegeben, wenn kein Papier anliegt, Papierende oder Farbbandende erreicht ist, außerdem leuchtet die Warnlampe dann noch auf. Der Summer läßt sich über leicht erreichbare DIP-Schalter abschalten. Eine sinnvolle Absicherung: Auch im abgeschalteten Zustand ertönt der Summer bei schwachen Batterien.

Mit einem weiteren DIP-Schalter läßt sich der Zeilenabstand von 1/6 auf 1/9 Zoll umstellen. Damit sind die hardwareseitigen Einstellmöglichkeiten aber schon vorbei. Alle Einstellungen, die das Schriftbild betreffen, müssen vom Computer übertragen werden.

Der Brother HR-5C wird vom Computer mit den gewohnten OPEN-Befehlen unter der Geräteadresse 4 aktiviert. Dabei kann durch die Wahl der Sekundäradresse entweder der Großschrift/Grafikmodus oder der Klein-/Großschriftmodus eingestellt werden. Auch der reverse Ausdruck ist möglich (CHR$(18)). Mit CHR$(14) werden Zeichen mit doppelter Breite ausgegeben. Eine Besonderheit ist die Möglichkeit, sich ein eigenes Zeichen definieren zu können. Ein Zeichen wird mit 7 DATA-Werten beschrieben. Auf die gleiche Art und Weise funktioniert das übrigens mit den Commodore Druckern VC 1526 und MPS 802. Durch diese Einzelnadelansteuerung müßte es auch möglich sein, sich hochauflösende Grafiken ausdrucken zu lassen, auch wenn das recht mühsam und langsam sein würde. Apropos Geschwindigkeit: Bei 30 Zeichen pro Sekunde braucht man schon einige Geduld. Diese Zeit wird noch erhöht durch die geringe Geschwindigkeit des Thermokopfes beim Zurückfahren zum Zeilenanfang und durch das Weiterspulen des Farbbandes nach jeder Zeile. Denn das Farbband ist ein Wegwerf-Band, es kann nur einmal verwendet werden.

Der Brother HR-5C ist besonders dort zu empfehlen, wo der Geldbeutel nicht allzu dick ist (Preis zirka 500 Mark) und wo sehr großer Wert auf geringe Geräuschentwicklung gelegt wird. Ein weiteres Plus ist die vollständige Kompatibilität zum Commodore-Zeichensatz. Eine Einschränkung stellt die Druckgeschwindigkeit dar und die fehlende Möglichkeit, normales gelochtes Endlospapier zu verwenden. Seine Druckmöglichkeiten sind sehr einfach. Sie beschränken sich auf Reversdruck und Breitschrift. Zusammengefaßt: der HR-5C ist ein einfacher Thermodrucker für einfache Anwendungen. (gk)



EIN STAR, DER ES IN SICH HAT

Drucker müssen schon einiges bieten, um konkurrenzfähig zu sein. Der Drucker Delta 10 von Star braucht sich dabei nicht zu verstecken.

Nach dem Auspacken des Druckers ist man zuerst einmal überrascht von der Anzahl der mitgelieferten Einzelteile. Das reichhaltige Zubehör findet man normalerweise nicht bei vergleichbaren Druckern. Abgesehen von den sonst auch üblichen Beigaben wie Farbband, Kunstoffdeckel und Handbuch erhält man auch noch zwei verschiedene, gleichzeitig verwendbare Papierführungen, einen Rollenhalter für Endlospapier auf Rolle und zusätzliche Reservesicherungen.

Im ersten Moment sieht das etwas verwirrend aus. Wohin mit den ganzen Teilen? Im Handbuch, wird man schnell fündig. Schon beim Durchblättern erkennt man an den Illustrationen, wie alles zusammengehört, die Beschreibungen sind fast unnötig, doch sehr exakt. Folgerichtig beginnt es bei der Montage der einzelnen Zubehörteile, beschreibt die Grundeinstellung des Gerätes, wie man sie ändert und erst danach die Möglichkeiten, das Druckbild zu verändern. Am Ende werden einige (der etwas klein geratenen) Tabellen erklärt und Beispiele zu den verschiedensten Druckbefehlen gegeben, so daß auch ein Unerfahrener auf diesem Gebiet nur selten Schwierigkeiten haben wird. Das Handbuch — es liegt in englisch und in deutsch vor — macht insgesamt einen guten Eindruck. Auch wenn es nicht mit einem Glanzumschlag und Ringbuchlochung aufwartet.

Der Delta 10 (Preis zirka 1750 Mark) kann gelochtes Endlospapier ebenso verarbeiten wie einzelne Blätter und Papier auf Rollen. Die zum Umstellen notwendigen Tätigkeiten sind problemlos, so etwa die Montage und Demontage der Tractoreinheit. Dieser Vorteil wird jedoch mit einem Nachteil erkauft: Eben weil die Tractoreinheit oben sitzt, also das Papier zieht und nicht schiebt, gibt es keine nützliche Papierabrißkante in der Nähe des Druckkopfes. Zum Abreißen eines Blattes muß es weitertransportiert werden bis zum Ende des Kunstoffdeckels. Daß ein Zurückdrehen des Papiers in Verbindung mit der Tractoreinheit nicht möglich ist, ist ärgerlich und kostet jedesmal ein unnötiges Blatt.

Zwei Schnittstellen

Selten findet man bei Druckern standardmäßig zwei Schnittstellen. Die Star-Familie besitzt denn auch eine Centronics- und eine RS232-Schnittstelle. Über DIP-Schalter können sie umgeschaltet werden. Der Delta 10 besitzt 3 verschiedene DIP-Schalter. Leider ist davon nur einer von außen zugänglich, die anderen beiden erreicht man nach Demontage des Gehäusedeckels.

Der Delta 10 ist mit einem Standard-Zeichengenerator (ROM) und einem freiladbaren Zeichengenerator (RAM) ausgestattet. Mit Hilfe des ladbaren Zeichengenerators ist es möglich, sich bis zu 192 Zeichen nach eigenem Wunsch zu generieren. Außerdem lassen sich durch Ansteuern der einzelnen Nadeln hochauflösende Grafiken erzeugen (Bit-Image-Graphicmode). Es existieren vier verschiedene Modi, von einfacher Dichte mit 480 Punkten bis hin zu 1920 Punkten pro Zeile.

Die Schriftarten

Der Drucker bringt je nach gewählter Schriftgröße 80,96,136 Zeichen und bei doppelter Breite 40,48 oder 68 Zeichen pro Zeile zu Papier. Als Schriftart kann man wählen zwischen dem normalen ASCII-Standard-Zeichensatz, Kursivschrift und 8 verschiedenen internationalen Zeichensätzen. Außerdem kann man Pica, Elite, komprimierte und gedehnte Schrift sowie Doppeldruck, uni- und bidirektionalen Druck einstellen. Auch ist es möglich, sämtliche Zeichen unterstrichen, hochgestellt (superscript) und tiefgestellt (subscript) darzustellen.

Es existieren einige Einstellmöglichkeiten zum vertikalen Zeilenvorschub. Der Zeilenabstand läßt sich bis zu einer Stufe von 1/144 Inch einstellen. Auch der Seitenvorschub läßt sich variieren. Man bestimmt die Länge des Papiers bis zur Perforation, kann einen Seitenvorschub kurz vor Ende des Blattes programmieren und die Position bestimmen, die die erste Zeile eines Blattes haben soll. Selbstverständlich lassen sich alle Einstellungen abschalten. Auch Tabulatoren lassen sich vertikal und horizontal einstellen, inklusive des linken und rechten Randes. Allein dafür gibt es zehn verschiedene Befehle, die keine Wünsche offen lassen. Mit diesen Kommandos läßt sich jede mögliche Art von Formularen ausfüllen.

Daß dieser Drucker voll grafikfähig ist, braucht eigentlich nicht besonders erwähnt zu werden. Das ist bei heutigen Matrixdruckern schon Standard. Der Ausdruck von Grafiken läßt sich mit einfachen Befehlen steuern. Sie erlauben den Ausdruck mit einfacher Dichte, doppelter Dichte, doppelter Dichte mit doppelter Geschwindigkeit und superhoher Dichte mit doppelter Geschwindigkeit.

Prädikat: sehr gut

Der Delta 10 ist ein Drucker, der das Prädikat »sehr gut« verdient, bezogen auf seine Druckfähigkeiten. Auch die Druckgeschwindigkeit und die Möglichkeit sowohl gelochtes Endlospapier als auch Endlospapier auf Rollen und Einzelblätter zu verwenden werden positiv gewertet. Abstriche müssen bei der Handhabung und bei dem Bedienungskomfort gemacht werden. Negativ fiel dabei der umständliche Papiertransport bei Verwendung der Tractor-Einheit (zurück nicht möglich) auf.

(gk)



Seikoshas Größter: Test GP-550A

Die Seikosha-Drucker stehen seit den Zeiten des schon fast legendären GP-80, der als VC-1515 Drucker zum VC 20 bekannt wurde, im dem Ruf von »Billigdruckern«, zwar grafikfähig, aber ansonsten mit wenig Vorzügen ausgestattet.

Der GP-550A wird diesem Ruf jedoch nur zum Teil gerecht, nämlich hinsichtlich des günstigen Preises. Ansonsten verfügt er über eine Reihe von Eigenschaften, die ihn auf eine höhere Stufe als seine bekannten kleinen Brüder GP-80 und GP-100 stellen.

Da wäre zunächst einmal die Schrift. Der GP-550A kann über Steuerzeichen insgesamt 18 verschiedene Schriftarten anwählen, unter anderem auch Elite, Script und Kursiv. Die Standardeinstellung ist Pica. Außerdem ist Breit- und Schmalschrift einstellbar. Auch Proportionalschrift ist möglich, was bei einem Drucker dieser Preisklasse durchaus nicht selbstverständlich ist.

Der GP-550A verfügt über 8 Drucknadeln, die Zeichen normalerweise in einer 8 x 9-Matrix zu Papier bringen. Daneben ist aber auch ein Schönschreibmodus vorgesehen, in dem die Punkte innerhalb einer 16 x 9- oder 16 x 12-Matrix gesetzt werden. Der Druckkopf wird dabei um einen halben Punktabstand vertikal versetzt, so daß insgesamt 16 sich überlappende Punkte gedruckt werden können. Durch diesen »Correspondence« genannten Schönschreibmodus ergibt sich ein Schriftbild ähnlich einer Schreibmaschine, da die einzelnen Druckpunkte so ineinander übergehen, daß sie auf den ersten Blick nicht mehr als solche zu erkennen sind. Dabei ist es noch möglich, wiederum verschiedene Schriftarten zu verwenden. Durch den Schönschreibmodus reduziert sich allerdings die ohnehin nicht allzu hohe Druckgeschwindigkeit von normalerweise 50 Zeichen pro Sekunde auf die Hälfte.

Wie alle Seikosha-Drucker, verfügt auch der GP-550A über eine Einzelnadelansteuerung. Der Umgang damit ist recht einfach. Nach Empfang eines bestimmten Steuerzeichens interpretiert der Drucker die weiteren ankommenden Zeichen als Grafik-Daten zur Nadelsteuerung. Jedem Bit eines Zeichens ist dabei eine Drucknadel zugeordnet. Ist das betreffende Bit in dem Zeichen gesetzt, dann wird auch die zugehörige Nadel gesetzt und erzeugt einen Punkt. So lassen sich problemlos Sonderzeichen und Symbole erzeugen, aber auch hochauflösende Grafik über eine größere Fläche ist möglich. Dabei können Grafik und Text auch in der gleichen Zeile beliebig gemischt werden.

Neben dem 8-Bit-Grafikmodus gibt es noch einen erweiterten 16-Bit-Grafikmodus, in dem die zusätzlichen acht Grafikpunkte wieder »zwischen« die anderen gelegt werden (wie im Schönschreibmodus). Damit lassen sich wirklich durchgezogene Linien erzeugen, die Grafik wirkt kompakter.

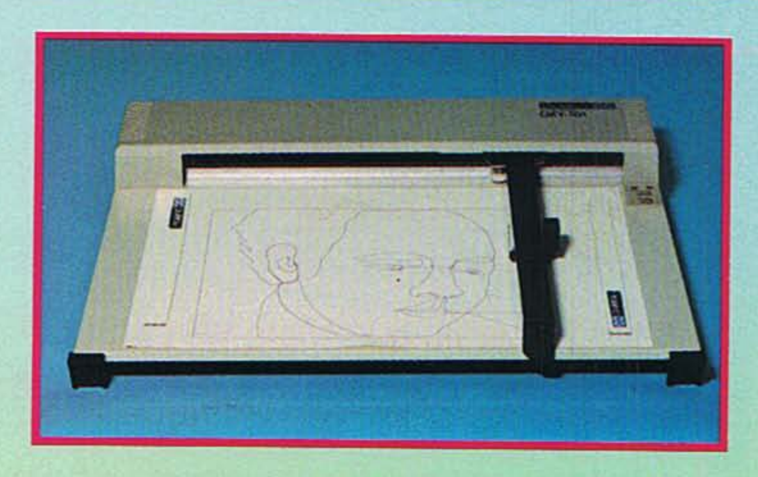
Der Seikosha GP-550A verfügt normalerweise über eine Centronics-Schnittstelle, ist jedoch auch (wie unser Testgerät) mit eingebautem VC-Interface lieferbar. Damit gibt es dann natürlich keine Anschlußprobleme an den C 64 oder VC 20: Einfach das Druckerkabel in den seriellen Port des Computers einstecken, einschalten, und schon ist die Anlage druckfertig. Damit erspart man sich unter Umständen eine Menge Ärger mit Interfaces und Treibersoftware.

Beim ersten Versuch, ein Programm zu LISTen, wird man allerdings feststellen, daß der GP-550A hinsichtlich des Zeichensatzes leider nicht Commodore-kompatibel ist. Es werden nämlich keine Steuer- oder Grafikzeichen gedruckt. Dafür sind aber die deutschen Umlaute möglich. Der GP-550A läßt sich nämlich per DIP-Schalter zwischen acht verschiedenen nationalen Zeichensätzen umschalten. Für Textverarbeitung und ähnliche Anwendungen wird man dabei sinnvollerweise den deutschen Zeichensatz wählen. Verzichtet man in seinen Programmen auf Grafiksymbole und schreibt Steuerzeichen als CHR$-Anweisungen, dann erhält man mit dem amerikanischen Zeichensatz recht brauchbare Listings.

Die praktische Arbeit mit diesem Drucker gestaltet sich einigermaßen angenehm, sieht man einmal von der niedrigen Druckgeschwindigkeit ab. Die Geräuschentwicklung ist für einen Seikosha erstaunlich zivil. Vom »Kreissägeneffekt« seiner kleinen Brüder ist nicht viel zu merken, obschon er von der Lautstärke her doch etwas über dem Niveau zum Beispiel eines Epson FX-80 liegt.

Papiereinzug und Farbbandwechsel sind mit wenigen Handgriffen und völlig problemlos erledigt. An Bedienelementen ist neben den üblichen Tasten für Zeilen- und Seitenvorschub eine zusätzliche Taste mit der Bezeichnung »Stop/Reset« vorhanden. Mit dieser Taste kann ein Druckvorgang unterbrochen und der Drucker wieder in den Einschaltzustand versetzt werden. Dadurch spart man sich das lästige Aus- und Einschalten des Druckers nach einem Fehler.

Insgesamt gesehen hat der Seikosha GP-550A im Test einen durchaus positiven Eindruck gemacht. Neben seinem günstigen Preis (1 098 Mark) besticht er vor allem durch seine Grafikfähigkeiten und durch die variablen Schriftarten. Nachteilig sind sicher seine vergleichsweise geringe Druckgeschwindigkeit und die fehlende Zeichensatz-Kompatibilität zum C 64. Den letzten Nachteil hat er aber mit den meisten Druckern seiner Leistungsklasse gemein. (ev)



Roland DXY-101 — ein Flachbettplotter im DIN-A3-Format

Die Entscheidung, ob Trommelplotter oder Flachbettplotter hängt vom Anwendungsfall ab. Der DXY-101 ist ein Flachbettplotter und besitzt nicht nur zwei Schnittstellen, sondern auch einen großen Befehlsvorrat.

Erfaßt ein Computer physikalische Meßwerte, so fallen jedesmal eine Unmenge an Zahlen an. Ein gewöhnlicher Drucker erstellt Zahlentabellen, die zumeist sehr unübersichtlich sind. Wesentlich besser erkennt man Zusammenhänge, wenn man sie optisch aufbereitet und in Kurven darstellt. Nun können auch Matrixdrucker Grafiken erstellen. Sie haben dabei jedoch einige gravierende Nachteile. Sie sind selten in der Lage, farbig zu drucken. Auch läßt sich das Papier nicht in beide Richtungen bewegen, eine Grafik kann dann nur von oben nach unten abgearbeitet werden. Ein dritter Punkt betrifft die Darstellung. Matrixdrucker setzen lediglich Punkte nebeneinander, während Plotter zusammenhängende Linien zeichnen. So gesehen, gibt es eine ganze Reihe von sinnvollen Anwendungsgebieten für Plotter.

Der Roland DXY-101 ist der kleinste Vertreter dieser Reihe im Roland-Programm. Außer dem Preis (zirka 2000 Mark) sind auch einige Fähigkeiten interessant. Als erstes fallen die beiden Schnittstellen an der Rückseite des flachen Plotters auf, eine Centronics- und eine RS232-Schnittstelle. Somit können Sie als 64'er Leser auch die im Heft 7 abgedruckte Treibersoftware hervorragend einsetzen.

Der DXY-101 wird mit einem Handbuch geliefert, das zwar alle möglichen Befehle beschreibt und auch zu jedem Befehl ein Beispiel enthält, jedoch in Englisch. Die abgedruckten Listings der kleinen Demonstrationsprogramme sind nicht für den C 64 geschrieben, lassen sich jedoch leicht anpassen. Die ebenfalls mitgelieferte Diskette enthält ein Demonstrationsprogramm, mit dem man sich jeden Befehl des Plotters einzeln vorführen lassen kann.

Ein großer Vorteil des Plotters ist sein Format. Mit DIN A3 läßt sich schon mehr anfangen als mit dem oft nicht ausreichenden Format A4. Das Papier, das möglichst glatt sein sollte, wird an beiden Seiten von zwei flexiblen Magnetstreifen gehalten. Es gibt keine Probleme beim Einlegen und Wechseln. Das gilt ebenso für die Zeichenstifte, die in verschiedenen Farben erhältlich sind. Die gelieferten Filzstifte empfanden wir jedoch als etwas zu dick. Außerdem verursachten sie ein gut wahrnehmbares Schleifgeräusch beim Plotten. Aber es gibt andere, auch feinere Stifte, die sich mit den mitgelieferten Spezialhalterungen einfach installieren lassen.

Der Befehlsvorrat des DXY-101 ist, gemessen an der Preislage, ausreichend groß. Neben den üblichen Befehlen zum absoluten und relativen Zeichnen und Bewegen ohne Zeichnen sind auch unterbrochene Linien möglich; der Abstand zwischen den Strichen ist frei wählbar. Auch x- und y-Achsen (mit und ohne Teilung) lassen sich mit einem Befehl erzeugen. Selbstverständlich können auch Buchstaben und Zahlen geplottet werden. Alle Zeichen können 16 verschiedene Größen annehmen, wobei nicht nur der Abstand zwischen ihnen, sondern auch die Schreibrichtung variabel ist. Zusätzlich gibt es noch einige Sonderzeichen, die zur zusätzlichen Begrenzung von Zeichnungen und Kurven eingesetzt werden können. Eine ganze Reihe von Befehlen befassen sich mit dem Zeichnen von Kreisen und Kreisausschnitten. Allerdings gibt es kein Kommando um Ellipsen zu zeichnen. Ganz auf selbst gestrickte Routinen kann man also nicht verzichten, will man den Plotter voll ausnutzen. Jedoch kann man sich eine kniffelige Aufgabe sparen: die Schraffur von Rechtecken. Auch sie kann mit nur einem Befehl ausgeführt werden.

Technische Daten

Der DXY-101 weist einen effektiven Plottbereich von 37 mal 26 cm auf, wobei der Stift sich mit einer Geschwindigkeit von bis zu 180 mm/sec bewegen kann. Seine Schrittweite beträgt 0,1 mm pro Schritt bei einer Wiederholgenauigkeit von 0,3 mm oder weniger. Die Positionierungsgenauigkeit liegt bei 1%.

Aufgrund seiner Größe ist der DXY-101 in der Lage, auch komplexere Zeichnungen aufs Papier zu bringen. Mit entsprechend feinen Stiften könnten allerdings feinere Linien gezogen werden. Die Filzstifte sind nicht so optimal. Die mechanische Stabilität der Führung ist für diese Preisklasse (um 2000 Mark) und Konstruktionsart durchaus ausreichend, auch wenn ab und zu leichte Resonanzen auftauchen können, die sich auf der Zeichnung durch leichtes Zittern bemerkbar machen. Sein Befehlsvorrat läßt nur wenig Wünsche offen und seine Schnittstellen lassen einen Anschluß an fast jeden Computer zu.

(gk)



SCHREIBMASCHINE – ANSCHLUSSFERTIG FÜR DEN C 64

Die Industrie merkt schon, wo der Hase hinläuft. Der C 64 mausert sich langsam zum Bürocomputer. Und für Bürocomputer, speziell für den Schriftverkehr im Büro, brauchen viele einen Drucker mit der Qualität einer Schreibmaschine.

Typenradschreibmaschinen sind in der Schriftqualität mit konventionellen Schreibmaschinen vergleichbar, weil auch sie für jedes Zeichen eine Type zur Verfügung halten. Diese Typen sind sternförmig auf dem Typenrad angeordnet, das sich leicht auswechseln läßt. Je nach Modell gibt es eine ganze Reihe von Typenrädern, für jede nur denkbare Schriftart ein anderes. Auch Räder mit Sonderzeichen (mathematische und andere Sonderzeichen, verschiedene Sprachen, etc.) gibt es. Typenraddrucker eignen sich auf Grund ihres guten Schriftbildes für Anwender, die optisch einwandfreie Schriftstücke verlangen.

Doch wir wollen in dieser Ausgabe nicht die Schreibmaschine als solche vorstellen, sondern den Anschluß der Schreibmaschine an den Computer, in diesem Fall der an den C 64.

Die Olympia Werke bieten die Compact 2 zwar als anschlußfertig für einen Computer an, jedoch nicht für den C 64 mit seiner seriellen Schnittstelle. Den Einbau einer seriellen Schnittstelle muß dann auch der Fachhandel vornehmen. In unserem Fall macht das für Olympia die Firma iti-Datentechnik in Leonberg. Sie baut eine Schnittstelle in das Gehäuse ein, so daß man lediglich das normale serielle Kabel, das auch zum Anschluß der Floppy VC 1541 dient, einstecken muß und schon ist die Verbindung hergestellt. Die Spannungsversorgung geschieht dabei über die Schreibmaschine. Intern hat die Compact 2 eine parallele Centronics-Schnittstelle, so daß auch bei einem eventuellen Systemwechsel eine andere Schnittstelle leicht eingebaut werden kann.

Das Interface ist, natürlich, Commodore-kompatibel. Das heißt, daß der Drucker über die Geräteadresse 4 angesprochen werden kann. Der C 64 sendet normalerweise kein Auto-Line-Feed Signal (das macht er nur, wenn man eine Primäradresse größer als 128 nimmt, also zum Beispiel OPEN 132,4). Um trotzdem mit »normalen» Primäradressen arbeiten zu können, muß nach dem Einschalten der Olympia die ON-LINE-Taste und die Halbzeilentaste gedrückt werden. Damit wird automatisch bei einem Wagenrücklauf ein Zeilenvorschub durchgeführt, solange der Drucker eingeschaltet ist.

Beim Einschalten des Gerätes kommt man in den Groß-/Kleinschrift-Modus, genauer gesagt den »Cursor-Down-Modus«. Durch Verwendung der Sekundäradresse 7 wird in den »Cursor-Up-Modus« geschaltet, das heißt Großbuchstaben werden als Kleinbuchstaben gedruckt und umgekehrt. Diese zwei Betriebsarten können ausgetauscht werden. Auch die Geräteadresse ist durch Trennen einer Brücke auf der Schnittstellenplatine auf 5 einstellbar.

Die Frage vor allem für uns Commodore-Besitzer ist natürlich, ob man sich auch Listings ausdrucken lassen kann. Vielleicht will sich dieser oder jener in seiner Freizeit Programme schreiben und möchte sich zum Ausdrucken nicht extra einen Matrixdrucker kaufen. Aber keine Sorge, das funktioniert ohne große Probleme. Man darf dabei aber nicht vergessen, daß ein Typenraddrucker nicht grafikfähig ist. Also Steuerzeichen können nicht dargestellt werden, wie man es gewohnt ist. Vergessen wurden sie jedoch auch nicht. Dieses Problem wurde so geregelt, daß vor jedem Steuerzeichen im Listing ein »!« vorangestellt wird, gefolgt von einem Buchstaben, der so auch im Listing am Bildschirm zu sehen ist, wenn man in den Groß-/Kleinschriftmodus schaltet. Das ist akzeptabel und eine gute Idee.

Viele Funktionen der Schreibmaschine können durch Einsatz von Steuersignalen auch vom Computer aus durchgeführt werden. Dazu gehören unter anderem der Hupton, die Zeilenschaltung, der Formularvorschub, der Wagenrücklauf, Ein- und Ausschalten der Sonderzeichenebene, mit der einige weitere Zeichen gedruckt werden können. Mit Sondersteuerzeichen (durch Senden verschiedener Escape-Signale) läßt sich die Schreibdichte (10,12,15 Zeichen/Zoll) umschalten, aber auch die Zeilenschaltung (1-,2-, 1½-zeilig) und die Halbzeile positiv und negativ (zum Hoch/Tiefstellen von Zeichen um eine halbe Zeile) einstellen.

Werden innerhalb von Strings Commodore-Steuerzeichen verwendet, so führen auch sie eine Funktion aus: CLR = Formfeed, HOME = Löschen des Schnittstellenspeichers (Puffer), CURSOR links = Leerschritt, CURSOR rechts = Rückschritt, RVS ON/OFF = Sonderzeichenebene ein-/ausschalten.

Die Olympia Compact 2 ist eine professionelle Schreibmaschine, die man nicht nur an den C 64 anschließen, sondern auch als reine elektronische Schreibmaschine einsetzen kann. In Verbindung mit dem C 64 und einer Floppy erhält man ein System, das den Weg zur wirklich professionellen Büroarbeit ebnet. Die leichte Handhabung und Bedienbarkeit der Compact 2 und der Preis von zirka 1500 Mark lassen dann auch nur wenig Wünsche offen. (gk)

Info: iti-Datentechnik, Telemannstraße 18, 7250 Leonberg, Tel.: 07152-6306

Marktübersicht:

Drucker für C 64/VC 20

Das Angebot an Druckern und Plottern für Homecomputer wird beinahe von Woche zu Woche umfangreicher. Diese Marktübersicht soll unseren Lesern helfen, sich einen Weg durch den »Druckerdschungel« zu bahnen.



Zwei Kategorien von Druckern und Plottern fehlen in dieser Übersicht. Das sind zum einen Geräte aus Preisklassen, die für den C 64-Anwender (und erst recht für den VC 20-Benutzer) nicht mehr interessant sind. Sie werden also nach Superdruckern für 5000 Mark ebenso vergeblich suchen wie nach DIN-A3-Plottern im fünfstelligen Preisbereich. Außerdem haben wir generell Drucker nicht berücksichtigt, die nur mit unvertretbar großem Aufwand an den C 64/VC 20 angeschlossen werden können.

Sie können also davon ausgehen, daß alle in dieser Marktübersicht vertetenen Drucker und Plotter sich ohne größere Probleme an den C 64/VC 20 anschließen lassen. Eine Reihe von Geräten ist sogar direkt über den seriellen Bus anschließbar. Bei diesen Geräten ist in der Rubrik »Interface« dann »C 64/VC 20« vermerkt.

Die Liste der direkt zum C 64/VC 20 kompatiblen Drucker ist wahrscheinlich nicht vollständig, da ständig neue Drucker mit bereits eingebautem VC-Interface auf den Markt kommen. An dieser Stelle sind daher auch die Anbieter entsprechend umgerüsteter Drucker aufgefordert, uns entsprechende Informationen zukommen zu lassen. Diese Marktübersicht soll in der nächsten Ausgabe noch um weitere, hier nicht berücksichtigte Geräte ergänzt werden.

Geräte mit Centronics-Schnittstelle setzen ein spezielles Interface und die entsprechende Treibersoftware voraus, um sie an den C 64/VC 20 anschließen zu können.

Entgegen einer unter Anfängern weit verbreiteten Meinung ist der serielle Port bei den Commodore-Computern nicht mit einer RS232C-Schnittstelle identisch. Allerdings sind im Betriebssystem schon die Routinen zur Verwaltung einer RS232C-Schnittstelle vorhanden, so daß ein solcher Anschluß (über den User-Port) nicht allzu schwer zu realisieren ist.

Ein wichtiger Punkt bei der Auswahl des Druckers ist der Zeichensatz. Nur Drucker, bei denen in der Rubrik »Zeichensatz« der Vermerk »Commodore« vorkommt, können tatsächlich ohne Schwierigkeiten den gesamten Commodore-Zeichensatz drucken. Geräte mit dem Vermerk »ASCII« können ebenso wie Typenraddrucker nur die Standard ASCII-Zeichen drucken, also keine Grafiksymbole oder Steuerzeichen. Ist der Drucker grafikfähig oder können Zeichen selbst definiert werden, dann lassen sich allerdings auch diese speziellen Zeichen per Software simulieren.

Zum Schluß sei noch gesagt, daß alle Preisangaben nur ungefähre Werte sind. Wer sich bei verschiedenen Anbietern informiert, kann unter Umständen günstiger einkaufen. (ev)

Anbieter-Liste Drucker & Plotter

Die hier aufgeführten Adressen sind vielfach keine direkten Bezugsquellen. Sie erhalten aber zumindest Datenblatt und Händlernachweis für den von Ihnen ins Auge gefaßten Drucker oder Plotter.

Adcomp Datensysteme, Olgastr. 15, 8000 München 19; Brother International GmbH, Im Rosengarten 14, 6368 Bad Vilbel; Canon Deutschland GmbH, Postfach 1209, 8033 München-Martinsried; Casio Computer, Kieler Str. 212, 2000 Hamburg 54; Citizen Vertrieb Deutschland, MVB, Brüder-Grimm-Str. 5, 6408 Ebersburg; C.Itoh Deutschland GmbH, Königsallee 21, 4000 Düsseldorf 1; Commodore Deutschland, Lyoner Str. 38, 6000 Frankfurt 71; Epson Deutschland GmbH, Am Seestern 24, 4000 Düsseldorf 11; Mannesmann Tally GmbH, Postfach 2969, 7900 Ulm; Micro Enterprises, Prinzregentenstr. 78, 8000 München 80; Mirwald Elektronik, Fasanenstr. 8, 8025 Unterhaching; NEC Europa, Wiesenstr. 148, 4040 Neuss; Neumüller GmbH, Eschenstr. 2, 8028 Taufkirchen; OKI Electric, Emanuel-Leutze-Str. 8, 4000 Düsseldorf 11; Olympia International, Postfach 960, 2940 Wilhelmshafen; Quen-Data GmbH/WELCO, Paul-Ehrlich-Str. 8, 6074 Rödermark; Robotron-Vertrieb, Unitronic GmbH, Münsterstr. 338; 4000 Düsseldorf 30; Seikosha-Vertrieb, Microscan GmbH, Postfach 601705; 2000 Hamburg 60; Star Europe GmbH, Frankfurter Allee 1-3, 6236 Eschborn; Synelec Datensysteme GmbH, Lindwurmstr. 117, 8000 München 2; Watanabe GmbH, Arzberger Str. 10, 8036 Hersching.

Modell	Typ	Zeichen pro Sekunde	Zeichen pro Zeile	Matrix	Farben	Papierart	Zeichensatz
Adcomp X 100	Trommelplotter	k.A.	k.A.	—	4	Endlos DIN A4	k.A.
BMC BX-80	Matrixdrucker	80	40/80/71/142	9 x 8	—	Endlos, Einzelblatt	8 regionale Zeichensätze
Brother EP-22	Thermodrucker	17	75	7 x 5	—	Thermopapier oder Normalpapier	ASCII plus Sonderzeichen
Brother HR-15	Typenrad	13	110—165	—	—	Endlos, Einzelblatt	je nach Typenrad
Canon A-1210	Tintenstrahl	40	80	—	—	Einzelblatt, Rolle	k.A.
Casio FP-101PL	Trommelplotter	11	8—80	—	4	Rolle	ASCII
Citizen 560 LG	Matrixdrucker	65	40	5 x 7	2	Rolle (Normalpapier 70 mm)	ASCII
Commodore 1526	Matrixdrucker	80	80	8 x 8	—	Endlos	Commodore
Commodore MPS-801	Matrixdrucker	50	80	7 x 6	—	Endlos	Commodore
Commodore VC 1520	Printer Plotter	12	10/20/40/80	—	4	Rolle (114 mm)	Commodore
Commodore MPS-802	Matrixdrucker	50	80	8 x 6	—	Endlos	Commodore
Epson FX-80	Matrixdrucker	160	48—137	9 x 11	—	Endlos, Einzelblatt, Rolle	9 nationale Zeichensätze
Epson EX-80	Matrixdrucker	100	40—137	9 x 9	—	Endlos	11 nationale Zeichensätze
Epson RX-80 (VC)	Matrixdrucker	100	40—137	9 x 9	—	Endlos	Commodore und 10 x national
Itoh 8510 A	Matrixdrucker	120	80—136	k.A.	—	Endlos, Einzelblatt, Rolle	ASCII
Itoh APY 1000	Flachbettplotter	k.A.	k.A.	—	8	Einzelblatt DIN A4	k.A.
Mannesmann MT 80	Matrixdrucker	80	40/66/80/132	8 x 7	—	Endlos, Einzelblatt	8 nationale Zeichensätze
Mannesmann PIXY 3	Flachbettplotter	k.A.	k.A.	—	3	Einzelblatt DIN A4	8 nationale Zeichensätze
Micro-Enterpr. ME 80	Matrixdrucker	80	40/80/142	9 x 8	—	Endlos, Einzelblatt	ASCII
Mirwald MP 1003	Flachbettplotter	k.A.	k.A.	—	4	Endlos, Einzelblatt DIN A3	ASCII
NEC PC 8023	Matrixdrucker	100	40—136	9 x 7	—	Endlos, Einzelblatt, Rolle	k.A.
Neumüller NPR-5500	Trommelplotter	k.A.	k.A.	—	4	Endlos, Einzelblätter DIN A4	ASCII
Oki Microline 80	Matrixdrucker	80	80/132	k.A.	—	Endlos, Einzelblatt, Rolle	ASCII
Olympia Compact 2	Typenrad	14	115—172	—	—	Endlos, Einzelblatt	je nach Typenrad
Quen-Data DMP 1180	Matrixdrucker	80	80—142	9 x 7	—	Endlos, Einzelblatt, Rolle	ASCII
Quen-Data DMP 81/80	Matrixdrucker	80	80—132	9 x 7	—	Endlos	ASCII
Robotron 6311	Matrixdrucker	100	80—120	9 x 7	—	Endlos, Einzelbatt, Rolle	ASCII
Robotron 6312	Matrixdrucker	100	132—190	9 x 7	—	Endlos, Einzelblatt, Rolle	ASCII
Robotron TD 40	Thermodrucker	40	40	—	—	Rolle (Thermopapier)	ASCII
Seikosha GP-100VC	Matrixdrucker	30	80	7 x 5	—	Endlos	Commodore
Seikosha GP-50A	Matrixdrucker	40	46	7 x 5	—	Rolle (95 mm)	ASCII
Seikosha GP-550A	Matrixdrucker	86	80/96/136	8 x 6	—	Endlos	ASCII
Seikosha GP-700A	Farbmatrixdrucker	50	80—160	8 x 7	7	Endlos	ASCII
Star Powertype	Typenrad	18	110/132/165	—	—	Endlos, Einzelblatt	je nach Typenrad
Star gemini-10X	Matrixdrucker	120	80/96/136	9 x 9	—	Endlos, Einzelblatt, Rolle	ASCII und Sonderzeichen
Star stx 80	Thermodrucker	60	80	5 x 9	—	Rolle (Thermopapier)	8 nationale Zeichensätze
Synelec CP-80	Matrixdrucker	80	80	8 x 7	—	Endlos, Einzelblatt	8 nationale Zeichensätze
Watanabe Miplot jun.	Flachbettplotter	k.A.	k.A.	—	4	DIN A4	8 nationale Zeichensätze

Pufferspeicher	Grafik	Interface	Bemerkungen	Preis ca.	Bezugsquelle
20 KByte	Kreise, Schriften, Achsen etc.	Centronics, RS232C, IEEE 488	2 Prozessoren, automatischer Stiftwechsel	2900,—	Fachhandel
—	Bit-Image Modus, 640 Punkte/Zeile	Centronics, RS232C optional	Bidrektionaldruck, Unterstreichen, 4 Schriftarten	1 136,—	Mirwald Electronic
2 KByte	—	RS232C, optional C 64/VC 20	Tastatur, LDC-Display, Rechenfunktion, als Schreibmaschine nutzbar	549,—	Fachhandel, Kaufhäuser
5 KByte	—	Centronics, RS232C	—	1 599,—	Fachhandel, Kaufhäuser
—	k.A.	Centronics	—	1 760,—	Schwind Datentechnik
—	Farbgrafik	k.A.	10 Zeichengrößen	699,—	Fachhandel, Kaufhäuser
—	Hochauflösende Grafik	C 64/VC 20, Centronics, RS232C, TTL, 20 mA	Schwarz-Rot-Druck, Vollzeichensatz	475,—	Fachhandel, Kaufhäuser
—	Blockgrafik	C 64/VC 20	Ein programmierbares Sonderzeichen, Bidirektionaldruck	995,—	Fachhandel, Kaufhäuser
80 Byte	Einzelnadelsteuerung (480 Punkte/Zeile)	C 64/VC 20	—	795,—	Fachhandel, Kaufhäuser
80 Byte	480 Plot-Positionen pro Zeile	C 64/VC 20	—	560,—	Fachhandel, Kaufhäuser
2 KByte	Blockgrafik	C 64/VC 20	Ein frei definierbares Zeichen möglich	860,—	Fachhandel
2 KByte	Bit Image Modus (480—1920 Punkte/Zeile)	Centronics, RS232C, IEEE 488	diverse Schriftarten, Bidirektionaldruck, selbstdefinierbarer Zeichensatz	1 495,—	Fachhandel, Kaufhäuser
—	Einzelnadelsteuerung (480—1920 Punkte/Zeile)	Centronics, optional C 64/VC 20	128 verschiedene Schriftarten, Bidirektionaldruck	1 198,—	Fachhandel, Kaufhäuser
2 KByte	Einzelnadelsteuerung (480—1920 Punkte/Zeile)	C 64/VC 20	Wie Epson RX-80, aber anschlußfertig an C 64/VC 20	1 398,—	Mirald Electronic
1,5 KByte	k.A.	Centronics, RS232C	—	1 760,—	AC Copy DTV GmbH
132 Byte	—	Centronics, RS232C	—	1 300,—	Fachhandel
2 KByte	Einzelnadelsteuerung 640 oder 1280 Punkte/Zeile	Centronics, optional RS232C	Bidirektionaldruck, variable Schriftbilder, Steuercodes Epson-kompatibel	1 140,—	Fachhandel, Kaufhäuser
k.A.	Kreise, Kurven, Achsen, Linien	Centronics, RS232C	griechischer und mathematischer Zeichensatz	2 100,—	Fachhandel
k.A.	Einzelnadelsteuerung	Centronics, optional RS232C	Bidirektionaldruck, Superscript	1 189,—	Fachhandel
1,3 KByte	Kreise, Achsen, Zeichen und Symbole	Centronics, optional C 64/VC 20	—	2690,—	Mirwald Elektronik
8 KByte	k.A.	Centronics	—	1 730,—	Schwind Datentechnik
4 KByte	Kreise, Histogramme, Marken	Centronics, RS232C	—	1 990,—	Neumüller GmbH
2 KByte	Blockgrafik	Centronics, RS232C	—	1 150,—	Fachhandel
2 KByte	—	Centronics, RS232C	—	1 446,—	Bürofachhandel, Kaufhäuser
2 KByte	k.A.	Centronics, RS232C	—	995,—	Neumüller GmbH
k.A.	k.A.	Centronics, RS232C	—	650,—	Neumüller GmbH
k.A.	k.A.	Centronics, RS232C	—	999,—	Unitronic GmbH
—	k.A.	Centronics, RS232C	—	1 199,—	Unitronic GmbH
80 Byte	k.A.	TTL	—	467,—	Unitronic GmbH
—	Einzelnadelsteuerung	C 64/VC 20	Direktanschluß an C 64/VC 20	798,—	Fachhandel, Kaufhäuser
—	Einzelnadelsteuerung (276 Punkte/Zeile)	Centronics	—	379,—	Fachhandel, Kaufhäuser
—	Einzelnadelsteuerung	Centronics	Schnelldruck, Korrespondenzdruck	1 098,—	Fachhandel, Kaufhäuser
—	Einzelnadelsteuerung, Farbe jedes Punktes wählbar	Centronics, optional C 64/VC 20	—	1 700,—	Fachhandel, Kaufhäuser
165 Byte	—	C 64/VC 20 (Option)	—	1 400,—	Fachhandel
816 Byte	Bit-Image (480—1929 Punkte/Zeile)	Centronics, optional Commodore, RS232C, IEEE 488	Blockgrafik, 96 programmierbare Zeichen, 8 Schriftarten, Bidirektionaldruck, optional 8 KByte Puffer	1 195,—	Fachhandel, Kaufhäuser
80 Byte	Blockgrafik	Centronics, RS232C	Bidirektionaldruck	595,—	Fachhandel
k.A.	Einzelnadelsteuerung	Centronics, optional RS232C	variable Schriftarten, Bidirektionaldruck	1 140,—	Synelec Datensysteme GmbH
1 KByte	Kreise, Kurven, Achsen, ASCII-Zeichen	Centronics, RS232C	—	1 850,—	Watanabe GmbH

So sieht der Output aus

Auf dieser Seite werden die verschiedenen Schriftbilder der getesteten Matrixdrucker sowie zwei Zeichnungen der beiden Plotter gegenübergestellt. Es konnten nicht alle Möglichkeiten von jedem Drucker oder Plotter gezeigt werden. Doch können Sie sich vielleicht auch so schon einen kleinen Eindruck machen.

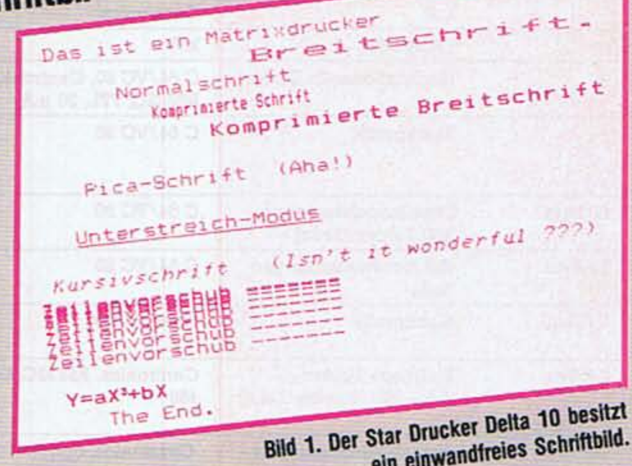


Bild 1. Der Star Drucker Delta 10 besitzt ein einwandfreies Schriftbild.

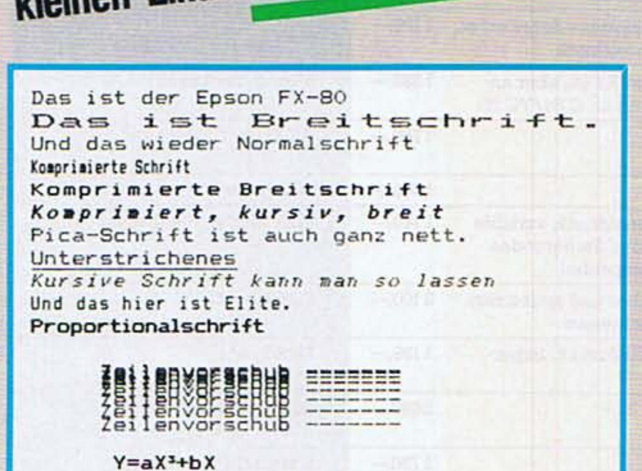


Bild 2. Epson FX 80, kein Unterschied zum Delta 10.

Das ist eine Schriftprobe des Seikosha GP-550A

Kursive Schrift

Kursiv- und Breitschrift

Unterstreichen ??? - Niemals !!!

Zusammengequetschte Schrift

Elite, Lauter tolle Schriftbilder

Und jetzt Superscript, was sonst ?!

Und anschliessend Subscript ist auch nicht schlecht.

▲ Bild 3. Der Seikosha GP 550 besitzt ein gutes Preis/Leistungs-Verhältnis.

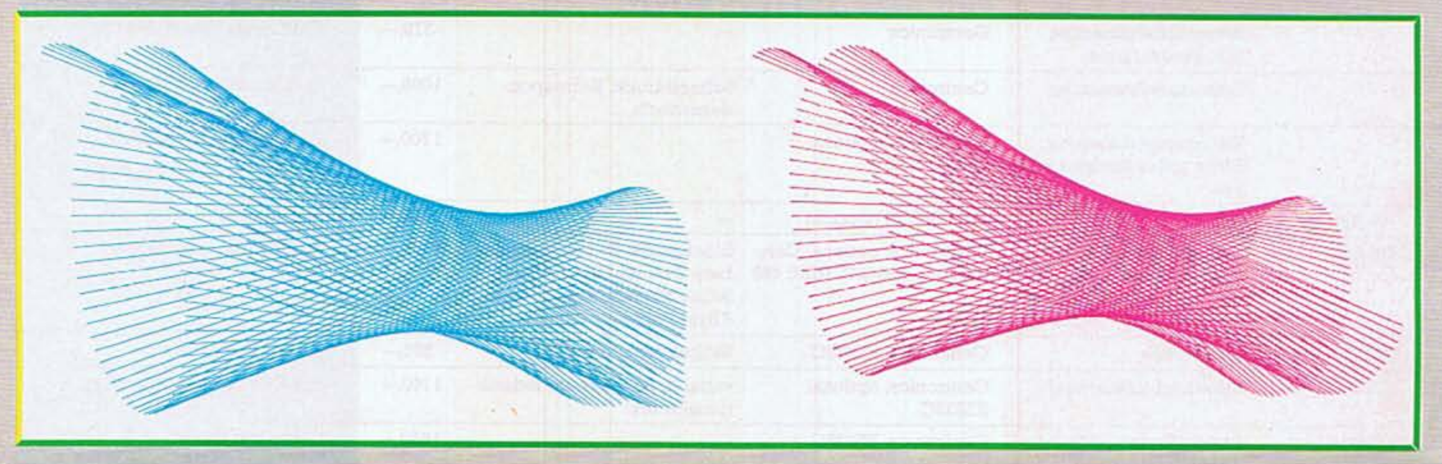
SCHRIFTPROBE COMMODORE-ZEICHEN:
MANFI WAS HERE. HEINI WAS NOT HERE

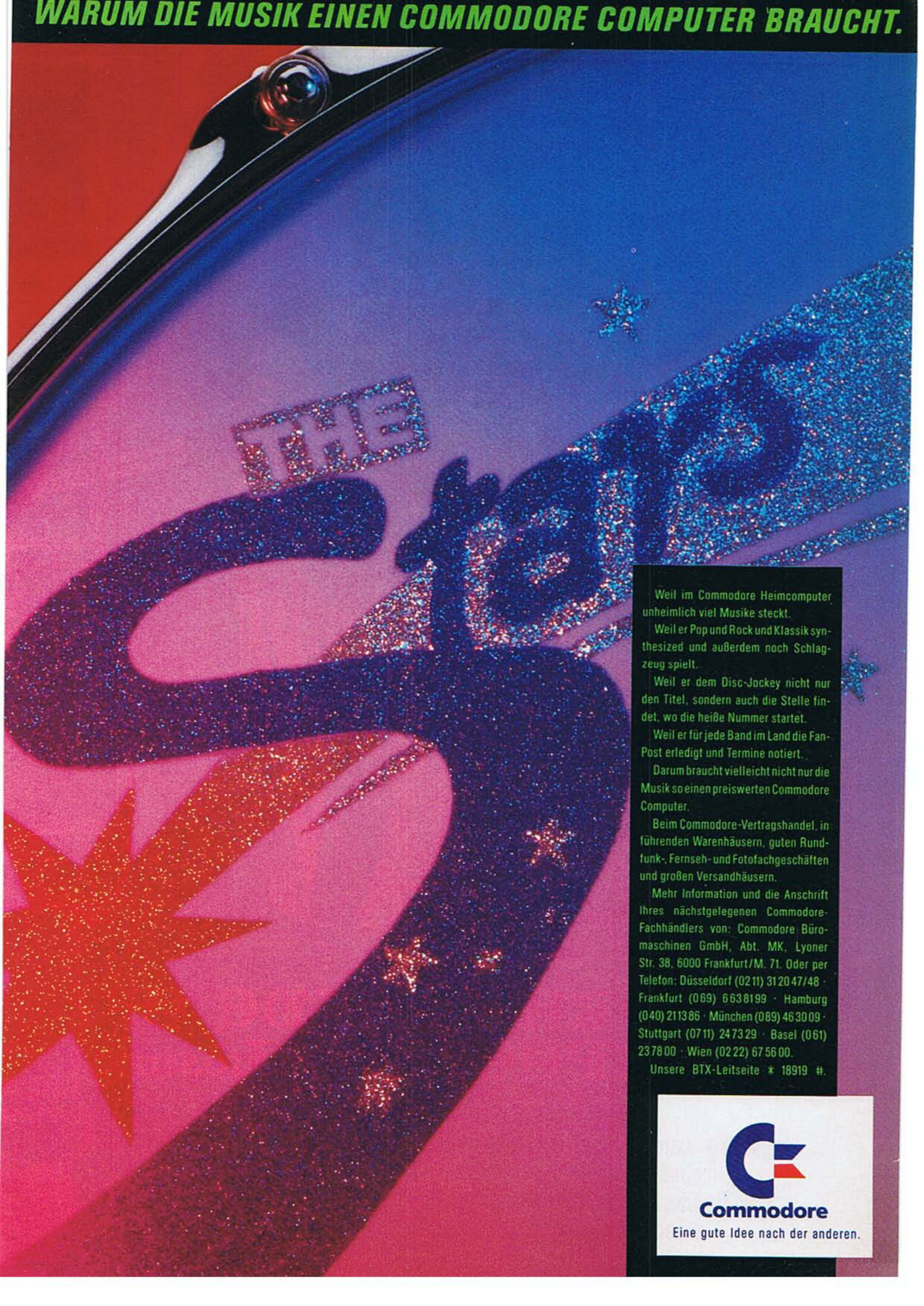
Schriftprobe des Brother HR-5C
Kleinschriftmodus DaDaDa-Aha-Aha-Aha

Reversschrift klein. REVERSSCHRIFT GROSS
Breitschrift. GANZ GROSS UND BREIT

◄ Bild 4. Alle Möglichkeiten des lautlosen Brother HR 5C.

Bild 5. und 6. Links der Adcomp X100 und rechts der Roland DXY-10.



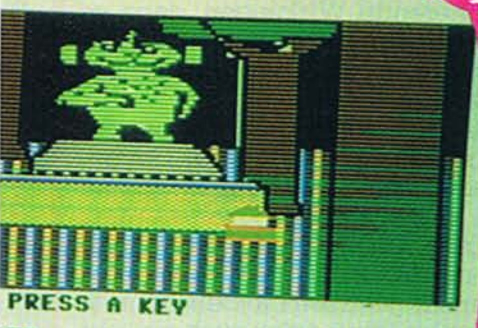


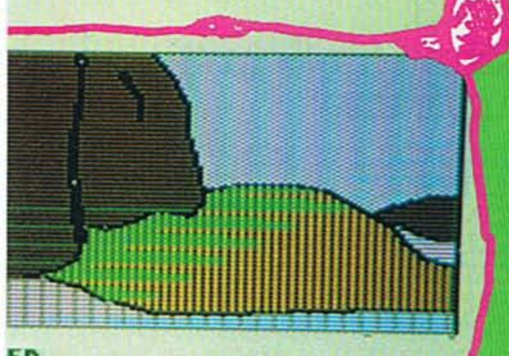
WARUM DIE MUSIK EINEN COMMODORE COMPUTER BRAUCHT.
THE Stars
Weil im Commodore Heimcomputer unheimlich viel Musike steckt.
Weil er Pop und Rock und Klassik synthesized und außerdem noch Schlagzeug spielt.
Weil er dem Disc-Jockey nicht nur den Titel, sondern auch die Stelle findet, wo die heiße Nummer startet.
Weil er für jede Band im Land die Fan-Post erledigt und Termine notiert.
Darum braucht vielleicht nicht nur die Musik so einen preiswerten Commodore Computer.
Beim Commodore-Vertragshandel, in führenden Warenhäusern, guten Rundfunk-, Fernseh- und Fotofachgeschäften und großen Versandhäusern.
Mehr Information und die Anschrift Ihres nächstgelegenen Commodore-Fachhändlers von: Commodore Büromaschinen GmbH, Abt. MK, Lyoner Str. 38, 6000 Frankfurt/M. 71. Oder per Telefon: Düsseldorf (0211) 312047/48 · Frankfurt (069) 6638199 · Hamburg (040) 211386 · München (089) 463009 · Stuttgart (0711) 247329 · Basel (061) 237800 · Wien (0222) 675600.
Unsere BTX-Leitseite * 18919 #.
Commodore
Eine gute Idee nach der anderen.

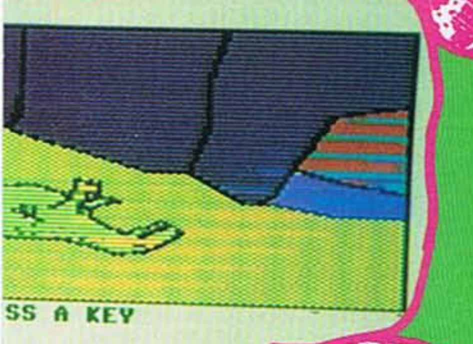


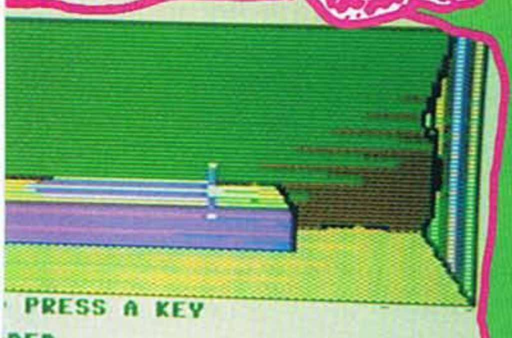
THE BLADE OF BLACKPOOLE – die Lösung

Im Gegensatz zu manchen Actionspielen, die schnell langweilig werden, sind Abenteuerspiele wie »The Blade of Blackpook« eine lohnende Anschaffung. Sie garantieren lang anhaltende Spielfreude, da die Aufgabe oft sehr schwierig ist. Für Computerbesitzer mit weniger Freizeit habe ich diese Lösung entworfen.









Die eigentliche Spannung eines Abenteuerspiels liegt natürlich darin, die Lösung selbst zu erarbeiten. Das dauert manchmal viele Monate, da einige der geforderten Lösungsschritte vollkommen unlogisch und nur durch reine Intuition (die Verwendung eines Monitors lehne ich ab) lösbar sind. So wird beispielsweise während der Suche nach dem Schwert Myraglym an einer Stelle verlangt, Bier in einen Fluß zu kippen. Wer dann keinen Maßkrug mit Bier aus einer Gaststätte bei sich hat, ist aufgeschmissen und muß von vorne beginnen.

Aus diesem Grund habe ich die Anleitung so gestaltet, daß im ersten Teil, der Hilfsliste, (Bild 1) nicht gleich alles verraten wird. Die Hilfsliste gibt lediglich Aufschluß darüber, wie ein spezielles Hindernis überwunden werden kann. Zum Auffinden der jeweiligen Position dient der zweite Teil, die Lagekarte (Bild 2). Jedes Feld ist mit einer Nummer und einem Buchstaben versehen. Sie geben einen Anhalt, unter welchem Punkt in der Hilfsliste nachgeschaut werden kann.

Der dritte Teil (Bild 3) besteht aus einer »Schritt-für-Schritt-Lösung«, mit der das Abenteuer innerhalb von zwei Stunden lösbar ist. Von der Verwendung dieses Teiles möchte ich dem ambitionierten Abenteurer eigentlich abraten, denn ist das Abenteuer erst gelöst, verliert das Spiel vollkommen seinen Reiz.

Vorab möchte ich allerdings noch einige allgemeine Hinweise zu »The Blade of Blackpool« geben: Die Aufgabe des Abenteurers besteht darin, das lange verschollene Schwert Myraglym zu finden und in eine Gaststätte nahe dem Ausgangspunkt zu bringen.

Erst dann ist das Abenteuer gelöst (500 von 500 möglichen Punkten). Dazu ist äußerst selten reine Gewalt

Teil 1. Die Hilfsliste

A	SWIMM E
B	Wer Honig hat, kann "CATCH BEE" sagen und Bienen für die Pflanze fangen
C	"TAKE BOTTLE". Damit verkleinert sich später das Boot
D	"BUY ALE": "TALK TO BARTENDER": TALK TO MEN"
E	BUY HONEY;ROPE;LAMP
F	"POUR POTION OVER BOAT". Damit schrumpft das Boot und kann mitgenommen werden.
G	"GIVE BEES TO THE PLANT"
H	"DROP HONEY". Sonst verhungert man kurz vor Schluß
I	"TIE POPE":"CLIMB UP"
J	"SING WITH BIRDS"
K	"LIGHT LAMP" Lampe muß ab da immer getragen werden, sonst geht sie aus.
L	"STEP ON PAD"
M	"TAKE TUNING FORK"
N	"TIE POPE":"CLIMB DOWN"
O	"DROP BOAT INTO WATER"
P	"POUR ALE INTO WATER"
Q	"PLACE BOOK ON ALTAR". Man erhält dann einen Schlüssel.
R	"HIT BOULDER WITH FORK"
S	Hier muß man so lange herumlaufen, bis man die "SCROLL" und den "LONG BOW" gefunden hat. Heraus geht es mit 4mal W oder 4mal E.
T	"READ PLAQUE"
U	"ROW DOWN"
V	"SAY MAHDEN" (steht in der Scroll) und bringt einen Pfeil: "SHOOT DRAGON"
W	"TAKE SWORD". Nicht Myraglym sagen!!!
X	"Wer das Schwert hat sagt: "PLACE SWORD ON ALTAR" und sagt dann: "SAY MYRAGLYM". Man landet dann mit dem Schwert in Bild 23
Y	Wer Hunger hat ißt jetzt, aber nur auf dem 2. Rückweg, unbedingt den Honig.

Bild 1. Mit dieser Hilfsliste und ein bißchen Erfahrung mit »The Blade of Blackpool« sollte man der Lösung näherkommen.

notwendig (was mir sehr gut gefällt). Wenn Waffen gebraucht werden, dann findet man sie, beziehungsweise erhält sie durch einen magischen Spruch. Das Spiel ist eigentlich nur mit viel unkonventionellem und einfallsreichem Denken zu lösen. Der »dann hau ich mich eben durch«-Typ wird vom Programm meist mit höhnischen Bemerkungen wie »we are a little violent today? abgespeist. Auch sonst sind die Bemerkungen oft mit Spott gewürzt, zum Beispiel:»you might fall and hurt yourself« beim Versuch einen Baum zu erklimmen. Die meist gegebene Antwort ist aber »that had no effect« und das hat mich manchmal fast zum Wahnsinn getrieben.

Abschließend seien hier noch ein paar Tips vermerkt:

1. Es wird viel gelaufen, also keinen Rückweg scheuen, beziehungsweise einen Weg zweimal gehen, um alle Sachen zu transportieren.
2. Vorsichtig mit allen Zauberwörtern umgehen. Ab einer bestimmten Spielstufe hat das Wort »Regnilo« verheerende Wirkungen, denn es werden alle Besitztümer wieder an ihren ursprünglichen Ort gezaubert.
3. Das Spiel hat kein Eigenleben (im Gegensatz zu »The Hobbit«). Gegenstände können beliebig weggelegt und später wieder geholt werden. Es gibt keine Kobolde, die Gegenstände stehlen.
4. Immer gelassen bleiben, denn es ist nur ein Spiel, wenn auch ein sehr schönes.

(Arnd Wängler/aa)

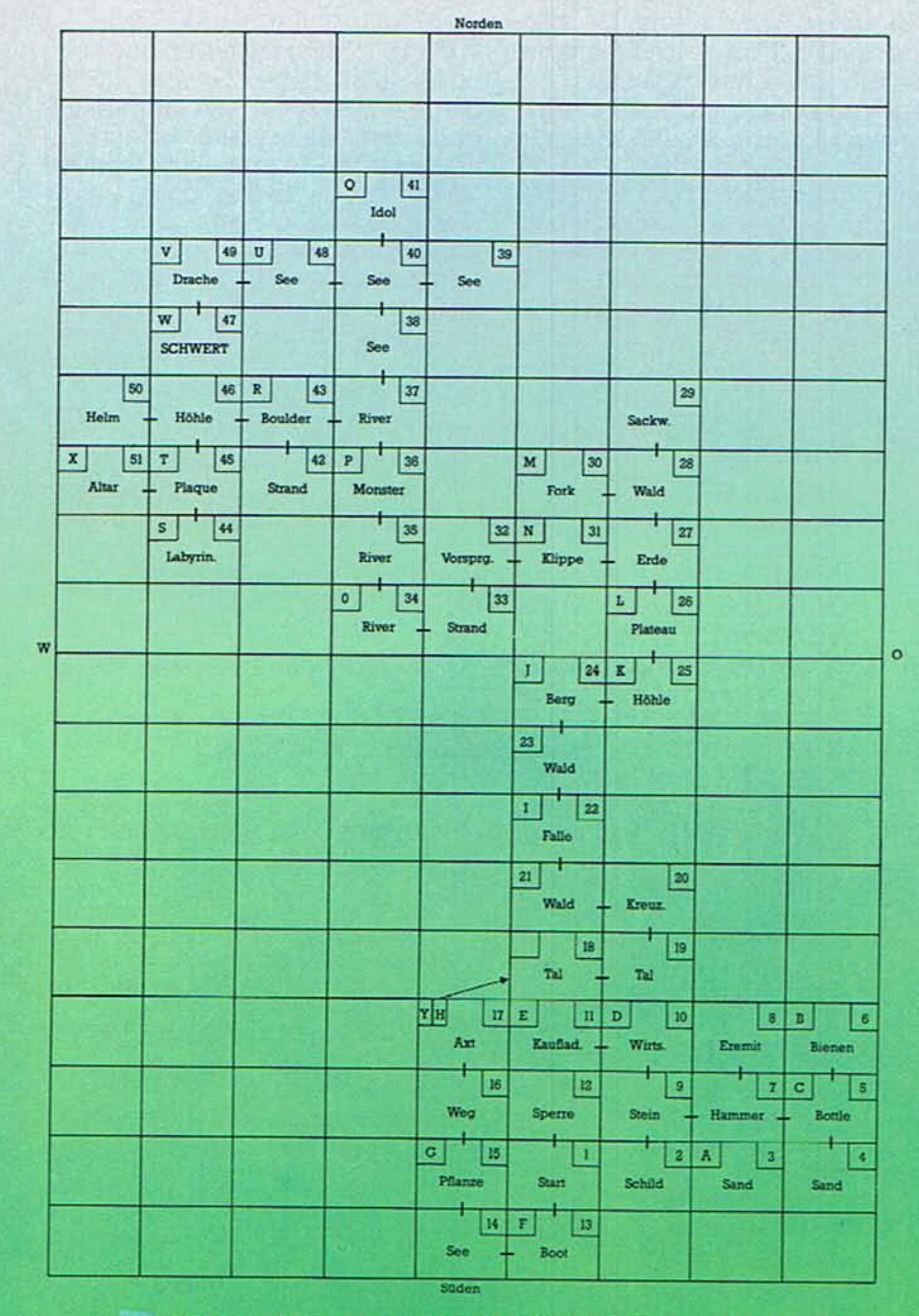


Bild 2. Der Lageplan von »The Blade of Blackpool«. Die Buchstaben in den Feldern beziehen sich auf die Angaben in Bild 1, die Zahlen auf Bild 3.

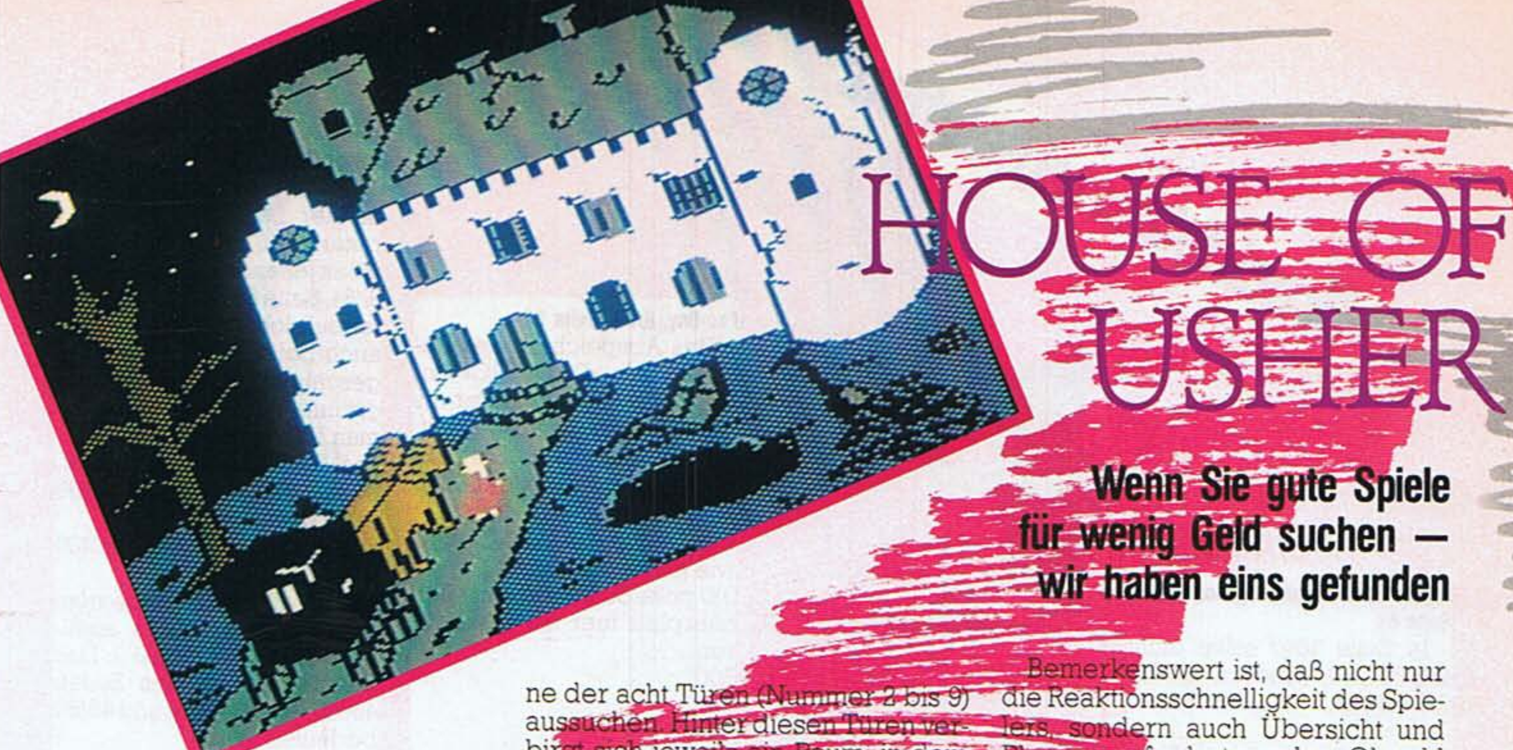
Teil 3. Schritt-für-Schritt-Anleitung

1-2-9-10 "TALK TO MEN" "TALK TO BARTENDER"-11"BUY ROPE;LAMP; HONEY"-10"BUY ALE""DROP BELT"-9-7-3"SWIMM E"-3-5-6 "CATCH BEE"-5"TAKE POTION" -4-3-2-1-15"GIVE BEES TO PLANT "-1-13"POUR POTION OVER BOAT""TAKE BOAT"-1-2"TAKE SHIELD"
-1-15-16-17"DROP HONEY"-18-19-20-21-22"TIE ROPE""CLIMB UP"-23-24-25
"LIGHT LAMP"-24-23"TAKE AMULET"-22-21-20-19-18-17-16-15-1-2-9-7-8-"GIVE
AMULET TO VOICE" "I"-7-9-2-1-15-16-17-18-19-20-21-22-23"TAKE ROPE"-24"SING WITH BIRDS"-26"STEP ON PAD"-27-31-"TIE ROPE" "CLIMB DOWN"-31-32-33-34"DROP BOAT" "-31"TIE ROPE""CLIMB UP"-27-28-30"TAKE
FORK"-28-27-31"TAKE ROPE" "TIE ROPE""CLIMB DOWN"31-32-33-34 "ENTER BOAT""ROW N"-35-36"POUR ALE INTO WATER"-37-38-39-41"PLACE BOOK ON ALTAR" "I"-40-38-37-43"HIT BOULDER WITH FORK"-46-45-44-44-44-44-..... "TAKE BOW" "TAKE SCROLL" "READ SCROLL"-45-46-43-37"ENTER BOAT"-38-40-48"ROW DOWN"49-"SAY MAHDEN""TAKE BOW""SHOOT DRAGON"-47"TAKE SWORD"-49-48-40-38-37-43-46-45-51 "PLACE SWORD ON ALTAR""SAY MYRAGLYM" man ist dann wieder auf Feld 23-22-21-20-19-18-17"EAT HONEY"-16-15-1-2-9-10

ENDE

Bild 3. Mit dieser Anleitung haben Sie »The Blade of Blackpool« in zwei Stunden gelöst.

die Lösung THE BLADE OF BLACKPOOLE



House of Usher

Wenn Sie gute Spiele für wenig Geld suchen — wir haben eins gefunden

Bei diesem Spiel aus dem Hause Kingsoft handelt es sich um ein Geschicklichkeitsspiel, bei dem der Titelheld die geheimnisvollen Rätsel des Hauses Usher lösen soll. Nach einem grafisch sehr schön gemachten Vorspann (Bild), in dem die Spielfigur das Haus betritt, kann er sich von der Empfangshalle aus eine der acht Türen (Nummer 2 bis 9) aussuchen. Hinter diesen Türen verbirgt sich jeweils ein Raum, in dem eine Aufgabe zu lösen ist.

Der letzte Raum vor der Schatzkammer (Nummer X) kann jedoch nur betreten werden, wenn vorher alle anderen Aufgaben gelöst wurden. Dabei muß der Titelheld sich vor Monstern retten, Abgründe und Falltüren überspringen, herabfallenden Felsen oder Kanonenkugeln ausweichen.

Bemerkenswert ist, daß nicht nur die Reaktionsschnelligkeit des Spielers, sondern auch Übersicht und Planung gefordert werden. Obwohl die Spielidee nicht ganz neu ist (Jumpman und Miner 2049er lassen grüßen), kann man sagen, daß »House of Usher« durch gute Grafik und Ton gefällt und lange Zeit interessant bleibt, da die verschiedenen Aufgaben nicht gerade einfach zu lösen sind. Das Spiel kostet nur 39 Mark auf Kassette oder Diskette.

(Manfred Kohlen/aa)

Fire-Galaxy

Man greife eine alte Idee auf, füge noch einige Schwierigkeitsgrade hinzu — fertig ist das neue Spiel. Es läuft auf dem VC 20 mit 16 KByte.

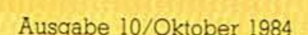
Bereits nach einigen Sekunden ist dem versierten Computerspieler klar, daß es sich um das altbekannte »Scramble« handelt. Allerdings wurde die Version Fire-Galaxy auf acht Spielstufen erweitert. Bei dem Spiel Scramble (Fire-Galaxy) dringt man möglichst tief in eine Höhle ein, in der die verschiedensten Feinde lauern. Diesen muß man ausweichen oder sie abschießen. Weiterhin ist zu beachten, daß man möglichst viele Treibstofftanks abschießt, um nicht infolge Treibstoffmangels abzustürzen. Dies wird natürlich immer schwieriger, je weiter man in die Höhle vordringt, denn man muß sich immer mehr darauf konzentrieren, den Angreifern auszuweichen.

Wer ein »Schlaftabletten-Reaktionsvermögen« besitzt, sollte sich entweder dieses Spiel nicht kaufen oder die F7-Taste mit einem Tesastreifen festkleben. Mit diesem Trick läßt sich das Spiel auf Zeitlupen-Tempo reduzieren. Wenn Sie schon beim Festkleben sind, dann können Sie auch den Feuerknopf am Joystick mit Tesafilm fixieren, denn diesen müssen Sie während der gesamten acht Spielstufen permanent drücken, was bereits nach kurzer Zeit schmerzhaft wird.

Auffallend an diesem Spiel ist nicht die Spielidee, sondern die wirklich ausgezeichnete Grafik und der gute Sound. Hier werden die grafischen und die musikalischen Fähigkeiten des VC 20 voll ausgeschöpft. Daß ein Spiel, das den VC 20 so gut nutzt und zudem ziemlich schnell ist, in Maschinensprache geschrieben wurde, versteht sich fast von selbst.

Fire-Galaxy von Kingsoft ist wirklich ein ausgezeichnetes Spiel, das sich jeder leisten kann. Denn es kostet (man glaubt es kaum!) nur 39 Mark.

(C. Q. Spitzner & B. Carti/ev)



Druckfehlerteufelchen.



Adreß- und Telefonregister, 5/84, Seite 64

In Zeile 1840 sollte nicht GOTO 610, sondern GOTO 510 stehen. (Gert Marx)

Abgerechnet wird mit dem C 64, 8/84, Seite 68

Mit Interesse habe ich den o.g. Artikel in Ihrer Zeitschrift gelesen. Da sich der Autor mit diesem Programm wegen der Thematik zwangsläufig an Gewerbetreibende wendet, fühle ich mich veranlaßt, einige Bemerkungen zu machen.

Im Text wird unter der Überschrift »Der feine Unterschied« auf die umsatzsteuerliche Behandlung von Portogebühren hingewiesen.

Die Deutsche Bundespost ist mit einem Teil ihrer Aufgaben (Postbeförderung) hoheitlich tätig und daher nicht Unternehmer im umsatzsteuerlichen Sinne, daher sind diese Umsätze der Post nicht steuerbar. Der gewerbliche Unternehmer ist — von Spezialfällen abgesehen — bei Weiterbelastung des Portos auch mit diesem Betrag umsatzsteuerpflichtig. Im Umsatzsteuerrecht wird die Portobelastung als Nebenleistung zur Hauptleistung (der Warenlieferung) gesehen und teilt deren Schicksal, das heißt der für die Lieferung geltende Steuersatz ist in der Regel auch für berechnete Portokosten zu erheben und an das Finanzamt abzuführen.

Im Interesse Ihrer gewerblichen Leser, für die eine Anwendung in Frage kommt, halte ich zur Vermeidung von leichtfertigen Steuerverkürzungen eine Richtigstellung für notwendig.

Zeile 60400 lautet:
...*10↑ur+.5))/10↑ur)

(Rainer Voß, Steuerberater)

Kurvendiskussion in hires, 7/84, Seite 118

Ich möchte hiermit auf einen von mir lang gesuchten Druckfehler aufmerksam machen. Er befindet sich auf Seite 118, Zeile 114.

Original:
```
114 SYS49629.Y+.5.X/8.
XAND255:SYS49744.2↑PEEK
(16722)
```
Richtig:
```
114 SYS49629,Y+.5,X/8,
XAND255:SYS49744,2↑PEKK
(16722)
```
(Thomas Lade)

Alle Tasten-, Zeichen- und Steuercodes, Teil 2, 5/84, Seite 107

Unter »Wo steht was?« hat sich ein Fehler eingeschlichen.

Unter Punkt (4) muß stehen: 60536 für 64632

Dasselbe gilt für Zeile 160.

64632 = $ FC78
60536 = $ EC78

Der Fehler ist also schon in der Programmerstellung entstanden, als sich für ein E ein F eingeschlichen hat

(Willi Heusser)

Pac-Boy, 8/84, Seite 89

Das Abspeichern des Datenblockes »DATA-PAC« auf Datasette hat sich leider als schwieriger herausgestellt, als angenommen. Ein Abspeichern ist aber dennoch folgendermaßen möglich:

Das Listing »DATA-PAC« wird wie folgt abgeändert...
```
310 poke 56,128:clr:rem speicherplatz fuer ma reservieren
500:
510:
520:
530:
540:
550:
560 for t=0 to 5937:read a :rem ma in Speicher bringen
570 poke 32768+t,a :next
```
Diese Änderung bewirkt, daß der Datenblock nun im Speicherbereich 32768 aufwärts abgelegt wird und weiterbearbeitet werden kann.

Nachdem sich DATA-PAC selbst gelöscht hat, ist folgendes Listing einzugeben...
```
0 data169,128,133,21,169,0,133,20,169,48,133,88,169,0,133,87,162,23,160
1 data0,177,20,145,87,136,208,249,230,21,230,88,202,16,240,169,0,133,251
2 data169,48,133,252,169,3,162,1,160,3,32,186,255,169,6,162,70,160,192,32
3 data189,255,169,251,162,0,160,72,32,216,255,96,77,65,45,80,65,67,0
4 for t= 0 to 76:reada:poke 49152+t,a:next
5 sys 49152
```
Startet man dieses Programm mit RUN, so wird das Programm »MA-PAC« auf Kassette abgespeichert. Damit das gesamte Programm ablauffähig ist, muß außerdem noch die Zeile 40 im Listing »PAC-BOY« zu
```
40........:load "ma-pac",1,1
```
abgeändert werden. Abschließend sei noch gesagt, daß die Programme »PAC-BOY« und »MA-PAC« auf der Kassette hintereinander abgespeichert werden sollten.

(H. Schlangmann)

Spring Vogel, 9/84, Seite 82

Das beste Listing das mir in meiner Laufbahn als Computerfreak untergekommen ist, ist jenes aus der Ausgabe 9/84, Seite 82. Das Druckfehlerteufelchen hat jedoch auch bei diesem Listing zugeschlagen. Damit das Programm korrekt läuft sollte man folgendes ändern:

Die Zeilennummern 59999, 60199, 60299, 60399, 60499 umändern in 60000, 60200, 60300, 60400 60500.

Nach Zeile 60030 sollte man noch folgende Zeile einfügen: 60100 REM BILD 2. Das Zeichen »_« in den Zeilen 46050, 48041, 49096 und 49160 bedeuted »—«.

Ansonsten stimmt alles beim »SPRING VOGEL«, und es wurde bei mir sofort zum Listing des Jahres gewählt!

(Lettner Gerhard)

Merge für C 64, 10/84, Seite 105

2. muß wohl richtig lauten:
```
POKE 43,(PEEK(45)+256*
PEEK(46)-2) AND 255
<RETURN>
POKE 44,(PEEK(45)+256*
PEEK(46)-2)/255
<RETURN>
```
dann funktioniert es.

(Klaus de Boor)

Index-sequentielle Datei, 9/84, Seite 56

Folgende Zeilen müssen korrekt lauten: 3350 IF R$="L" THEN GOSUB 4000: GOTO 3220
3340 IF R$="A" THEN GOSUB 12000: GOTO 3240.

Das macht den Kleinen größer, 9/84, Seite 112

Die beiden Bildunterschriften wurden vertauscht.

Der Commodore 64 wird zum PC – mit Vizawrite 64

Das Textverarbeitungsprogramm »Vizawrite 64« ist mit Leistungsmerkmalen ausgestattet, die sehr nahe an Wordstar herankommen. Dabei kosten Vizawrite und Computer zusammen weniger als ein vergleichbares Programm für einen Personal Computer.

Sie haben sich sicherlich schon gefragt, worin der Unterschied zwischen einem »Heimcomputer« wie dem C 64 und einem Personal Computer wie dem IBM-PC liegt. Das Computerlexikon definiert einen Heimcomputer als einen preislich zwar günstigen, aber wenig leistungsfähigen Computer für Heim- und Hobbyanwendungen. Ein Personal Computer ist in diesem Lexikon als ein hauptsächlich am Preis (5000 bis 15 000 Mark) und der fast unbegrenzten Erweiterungsfähigkeit erkennbarer Computer beschrieben.

Rein technisch gesehen ist der Unterschied gar nicht so frappierend. Ein Vergleich fällt in manchen Bereichen (Grafik, Farbe, Ton) nicht selten zugunsten des Heimcomputers aus. Derartige Zusatzfunktionen sind beim PC, wenn überhaupt, oft nur mit sehr viel Geld zu realisieren. Anders sieht es bei der Verfügbarkeit sehr leistungsfähiger Anwenderprogramme aus, hier haben die PCs die Nase vorn. Ob Datenbanken (dBase II), Textverarbeitung (Wordstar, Word) oder Tabellenkalkulation (Multiplan), jedes dieser Programme ist ein Spitzenprodukt. Mit »Vizawrite 64« wurde ein Programm geschrieben, das den Abstand zwischen einem »Heimcomputer« und den PCs weiter verringert. Der Preisvorteil der Heimcomputer ist geblieben.

»Vizawrite 64« ist mit der lange erwarteten deutschen Version in der Lage, den deutschen Zeichensatz auf dem Bildschirm darzustellen. Eine unabdingbare Forderung an ein Textverarbeitungsprogramm für den deutschen Markt wurde damit erfüllt. Die deutsche Version ist als Modul und auf Diskette erhältlich. Beide Versionen arbeiten mit einem oder zwei Laufwerken und sogar der Datasette zusammen. Das deutsche Handbuch (56 Seiten) macht insgesamt einen guten Eindruck und beschreibt ausführlich die enormen Fähigkeiten dieses Programms.

Diese Fähigkeiten wollen wir nun etwas genauer betrachten. Nach dem Laden (beziehungsweise Einschalten, bei der Modulversion) erscheint das Hauptmenü (Bild 1). Hier hat man die Auswahl zwischen dem Erstellen einer neuen Textdatei, dem Laden einer bestehenden Textdatei oder des Disketteninhalts und dem Senden von DOS-Befehlen (New, Scratch ...). Vom Hauptmenü aus wird auch, nach getaner Arbeit, das Programm verlassen. Zum Anwählen der einzelnen Aktivitäten dienen die Funktionstasten.

```
Vizawrite        Word Processing System

Select Activity:-

F1   -   Edit Old Document
F3   -   Create New Document
F5   -   Directory
F7   -   Disk Commands
F8   -   End Of Session

-Press Required Function Key.-
```

Bild 1. Das »noch« englische Hauptmenü

Zum Schreiben einer neuen Textdatei oder nach dem Laden eines bestehenden Dokuments wird in den Textverarbeitungsmodus geschaltet (Bild 2).

```
Vizawrite               name:Viza Hardcopy
Un Page   1 of   1      Line  20 Col     1
    Dies ist der Textverarbeitungsmodus
    Es sind 20 Zeilen und bis zu 240
    Spalten möglich.

    Von hier aus werden auch alle Format
    und Controllbefehle eingegeben.
    Dazu dient in besonderem Maße die
    Formatzeile direkt über diesem Text

    Mit Vizawrite 64 können bis zu 34000
    Zeichen auf einmal in den Computer
    eingegeben werden. Der Computer wird
    dabei kaum merklich langsamer.

    Alle Befehle sind einfach zu hand-
    haben und schnell zu erlernen. Im
    Notfall hilft das deutsche Handbuch
    oder das "Hilfe" Textfile, das auf
    der Diskette mitgeliefert wird.
(End of Page)
```

Bild 2. Der Textverarbeitungsmodus

Das zur Verfügung stehende Arbeitsfeld ist 20 Zeilen hoch und hat eine variable Breite von bis zu 240 Spalten. Auf dem Bildschirm erscheinen aber immer nur 40 Zeichen. Ab dem 38. Zeichen wird der Text horizontal verschoben (scrolling). Besitzer eines PCs kontern hier: »Aber mein PC kann jederzeit 80 Zeichen pro Zeile darstellen.« Stimmt, aber wenn man bedenkt, daß selbst Wordstar in der Regel nur 65 Zeichen benutzt, ist dies nicht unbedingt ein Nachteil. Auch hat nicht jeder Computerbesitzer einen Monitor, der die notwendige Auflösung für 80 Zeichen bereitstellen kann, sondern einen ganz normalen Fernseher. Für diesen sind aber 40 Zeichen pro Zeile mit horizontalem Verschieben geradezu ideal. Daß der ganze Text trotzdem, ohne lästiges Hin- und Herfahren mit dem Cursor überschaubar ist, ermöglicht die »WIDTH = Bildschirmbreite«-Funktion. Auf Tastendruck (CBM + w) wird der gesamte Text auf 38 Zeichen umformatiert. Der Zeitbedarf für diese Umformung ist fast nicht meßbar.

Alle Informationen des Textverarbeitungsmodus stehen in den oberen drei und der untersten Zeile. Dort erscheinen Fehler-, Systemmeldungen, die Seiten-, Zeilen- und Spaltennummern sowie der Name des Dokuments. Die vierte Zeile ist die sogenannte Formatzeile. Sie bestimmt das Aussehen des Textes. Hier werden Textbreite, Tabulatoren und die später noch beschriebenen Kontroll-Marken festgelegt. Die Formatzeile ist an jede beliebige Stelle im Text kopierbar (CTRL + f). Sie gibt dem Text ab dort ein neues Aussehen.

»Vizawrite 64« ist ein bildschirmorientiertes Textverarbeitungsprogramm. Alle Texte werden formatiert eingegeben und erscheinen praktisch so auf dem Bildschirm, wie sie später auf dem Papier ste-

hen. Im Gegensatz zu anderen Textverarbeitungsprogrammen wird kein Wort am Zeilenende abgeschnitten, sondern statt dessen in die nächste Zeile übernommen (word wrap). Jedes Dokument kann beliebig viele Seiten haben, die einer späteren Druckseite entsprechen. Bei einem freien Arbeitsspeicher von 34 000 Zeichen finden so zehn »normal« beschriebene Schreibmaschinenseiten Platz. Das ist wesentlich mehr als bei anderen Programmen.

Schnelles Arbeiten möglich

Vizawrite ist schnell. Wir haben den Test gemacht und den gesamten Speicher gefüllt. Dabei gab es keine Probleme mit der Geschwindigkeit. Selbst das letzte Zeichen wird ohne Verzögerung angenommen. Beachtlich gering ist auch der Zeitverbrauch beim Umformatieren. Von 40 auf 80 Zeichen haben wir nur eine Sekunde gemessen. Der IBM-PC braucht dafür, zusammen mit Wordstar, mehrere Minuten (je nach Länge des Textes).

Die Befehle des Textverarbeitungsmodus bestehen im wesentlichen aus drei Hauptgruppen:
1. Die Systembefehle wie Laden, Finden, Drucken und vieles mehr.
2. Die Formatsymbole wie Tabulatoren und Befehle zur Druckersteuerung.
3. Die Funktionstasten.

Alle Befehle und Funktionen von »Vizawrite 64« können zu jeder Zeit und an jeder Stelle des Textes aufgerufen werden. Dazu ist kein langwieriger Marsch durch verschiedene Menüs notwendig. Die Befehle sind einprägsam und logisch. So bedeutet beispielsweise CBM + s (s für SAVE), daß der Text abgespeichert wird. Obwohl die Befehle schnell erlernbar sind, kann man sich in der Anfangszeit mit einem Trick behelfen: Das mitgelieferte Programm »Hilfe« wird gemäß der Anleitung in die sogenannte »work page« oder Arbeitsseite geladen. Die Arbeitsseite ist ein separater Textspeicher, aus dem oder in den Texte kopiert werden. Alles, was der Anwender dann noch wissen muß, ist der Befehl CBM + g. Beantwortet er die Frage nach der anzuzeigenden Seite mit »w«, so befindet er sich auf der »work page«. Dort kann er sich dann über alle Befehle und Funktionen informieren.

Die Systembefehle sind die leistungsfähigsten Anweisungen von »Vizawrite 64«. Sie rufen wichtige Funktionen auf und beginnen immer mit der CBM-Taste. Daraufhin erscheint in der Kommandozeile die Frage nach der gewünschten Funktion.

Der Anwender hat nun die Auswahl zwischen 16 verschiedenen Funktionen:

1. CBM + c: Text kopieren. Mit Hilfe der Cursortasten wird der Text, der kopiert werden soll, markiert. Dies geschieht durch einfaches Darüberfahren mit dem Cursor. Die markierten Textteile erscheinen farblich abgehoben. Durch Abwärtsbewegen des Cursors werden ganze Zeilen, durch Drücken der f1-Taste ganze Seiten und mit der f3-Taste eine Bildschirmseite markiert. Nach dem Markieren wird nur noch der Bestimmungsort des Textteils festgelegt und mit Return quittiert. Der Text befindet sich dann an der neuen und der alten Stelle im Dokument.
2. CBM + m: Text verschieben. Hat die gleiche Funktion wie Text kopieren, mit dem Unterschied, daß der markierte Bereich an seiner ursprünglichen Position gelöscht wird.
3. CBM + f. Finden eines bestimmten Ausdrucks. Die Länge des Ausdrucks ist dabei auf 25 Zeichen beschränkt.
4. CBM + F: Finden eines Ausdrucks, ohne dabei auf Groß/Kleinschreibung zu achten.
5. CBM + g: Zu einer bestimmten Seite springen. Mit dieser Funktion erscheint die angewählte Seite im Textfenster. Andererseits werden so die separaten Seiten für Kopfzeile, Schlußzeile sowie die Arbeitsseite (work page) erreicht.
6. CBM + d: DOS-Befehle senden. Dieser Befehl dient der Handhabung des Diskettenlaufwerks.
7. CBM + M: Einfügen von Dokumenten. Eine sehr wichtige Funktion, mit der an einen Text im Speicher ein Textteil von Diskette nachgeladen wird. Der Merge-Befehl dient aber auch zum Einlesen verschiedener Dateiarten. Es können Dateien von anderen Textverarbeitungsprogrammen wie Wordpro 3+, Easyscript, SM-Text und vielen anderen eingelesen werden. Im Rahmen des später beschriebenen Erstellens von Serienbriefen dient diese Funktion dem Einlesen der Adreßdateien von Vizastar oder anderen Datenbanken.
8. CBM + n: Neubenennen eines Textes.
9. CBM + p: Drucken. In dem dann erscheinenden Druckermenü (Bild 3) können alle wesentlichen Einstellungen, die auf das gesamte Schriftstück wirken sollen, definiert werden.
10. CBM + q: Arbeit beenden. Mit dieser Funktion wird zum Hauptmenü (Bild 1) zurückgekehrt. Sollte der Text im Speicher mit der zuletzt geladenen Version nicht übereinstimmen, wird der Anwender gewarnt und zum Abspeichern aufgefordert.
11. CBM + r: Ersetzen eines Ausdrucks. Der auszutauschende Text wird wie unter 1. beschrieben markiert und durch bis zu 28 neue Zeichen ersetzt.
12. CBM + R: Globales Austauschen. Der Ausdruck wird nicht nur an einer bestimmten Stelle ersetzt, sondern im gesamten Text.



Bild 3. Das umfangreiche Druckermenü von Vizawrite 64

13. CBM + s: Abspeichern des Textes. Vizawrite produziert PRG-Files, in denen alle Informationen des bestimmten Textes mit abgelegt sind. Sowohl die Farbgebung, alle Formatierungen als auch die gewählte Druckereinstellung werden zusammen mit dem Text abgespeichert. Bereits bestehende Dateien mit dem gleichen Namen werden allerdings überschrieben.
14. CBM + t: Änderung der Farbgebung.
15. CBM + w: Ändern der Dokumentbreite auf 38 Zeichen.
16. CBM + x: Einstellen auf andere Gerätekonfigurationen. Bis zu zwei Diskettenlaufwerke sind ansprechbar, aber auch eine Einstellung auf die Datasette ist machbar. Wer seinen Drucker (oder mehrere Drucker) betreiben möchte, hat hier Gelegenheit, die Geräteadresse des Druckers auf 5 einzustellen. Wir haben es mit dem 1520 Printer/Plotter versucht und hatten Erfolg. Auch auf diesem Gerät wird jeder Text ausgedruckt (Bild 4). Die einzelnen Schriftarten des Plotters eignen sich besonders für Aufkleber und Beschriftungen.

Kein anderes mir bekanntes Text-

verarbeitungsprogramm hat eine ähnliche Fülle von Befehlen, die mit vergleichbarer Geschwindigkeit und Zuverlässigkeit funktionieren. Damit es auch bei der Texteingabe an keinem Komfort mangelt, haben einige Tasten Sonderfunktionen.

Die Funktionstasten dienen zum seitenweisen oder bildschirmweisen Vor- oder Zurückblättern sowie dem komfortablen Einfügen und Löschen. Ein Druck auf die f7-Taste genügt, um beliebig viele Zeilen in einen Text einfügen zu können. Ein zweiter Druck auf dieselbe Taste oder die Stop-Taste fügt den restlichen Text nahtlos wieder an. Die f8-Taste sorgt für das schnelle Löschen von Textteilen. Der Text wird, wie unter Punkt 1 der Befehlsfunktionen beschrieben, markiert und nach dem Quittieren mit der Return-Taste blitzschnell gelöscht. Nicht selten wird es bei der Texteingabe notwendig, große Sprünge durch den Text zu machen. Die Taste »HOME« bewirkt beim ersten Druck einen Sprung zum oberen Rand des Bildschirms und beim zweiten Druck den Sprung zum Anfang des Textes. Zurück geht es mit der »CLR«-Taste, die den Cursor an das Textende bewegt.

»Vizawrite 64« ist, wie schon gesagt, bildschirmorientiert. Deshalb ist es natürlich gut, schon bei der Eingabe auf die entsprechende Formatierung des Textes zu achten. Der Anwender wird dabei tatkräftig unterstützt. Seine Hilfsmittel sind Texttabulatoren, numerische Tabulatoren, Formatzeilen, Einrück- und Zentrierbefehle. Die f5- und f6-Tasten dienen dabei dem schnellen Sprung von einem Tabulator zum nächsten oder vom Anfang einer Zeile zum Ende. Auch die Kommandos zur Druckersteuerung bildet »Vizawrite 64« auf dem Bildschirm ab. Dabei sind eine Reihe von Kontrollbefehlen bereits »fest verdrahtet«. Fettschrift, Unterstreichen, Sub- und Superscript haben ihre festen Symbole, die mit vorangestellter CTRL-Taste in den Text eingefügt werden. Um aber optimale Flexibilität für die Fähigkeiten verschiedener Drucker bereitzustellen, kann jeder Anwender beliebig viele Befehle selbst definieren. Dies geschieht in der Formatzeile. Die Ziffern 0 bis 9 erhalten dort Werte, die, bei Wiederholung im Text, den Drucker steuern. Die bei manchen Druckern üblichen ESC-Sequenzen müssen dabei nicht explizit definiert werden. Es genügt, hinter der Zahl des Kontrollbefehls den Buchstaben der betreffenden Sequenz anzufügen.

```
Vizawrite 64 arbeitet auch mit dem 1520 Color Plotter zusammen

Wenn der Plotter auf Geraeteadresse 5 umgeschaltet ist, kann Vizawrite
64 sogar diesen Drucker ansprechen. Mit Textverarbeitung hat das
allerdings wenig zu tun. Aber auf diese Weise koennen Aufkleber,
farbige Beschriftungen und aehnliches auf einfache Weise realisiert
werden.

Durch die variablen Schriftbreiten des Plotters sind so enorme
Textmengen auf winzigstem Raum unterzubringen. Der Befehl dafuer
lautet: CBM x und danach P. Alle Einstellungen bezueglich der
Schriftgroesse sind allerdings vor dem Laden von Vizawrite an den
Plotter zu uebermitteln (siehe Handbuch).

Die Umlaute sind auch hier, wie bei allen Commodore Druckern, nicht
verfuegbar und muessen durch die Umschreibungen ersetzt werden. Andere
Funktionen, wie beispielsweise der Blocksatz bleiben, wie sie sehen
voll erhalten
```

Bild 4. Auch der 1520 Printer/Plotter läßt sich verwenden

Optimale Druckersteuerung

Das Umschalten eines Epson-Druckers auf »Elite«-Schrift sieht in der Formatzeile beispielsweise so aus: 0 = M. Da aber beliebig viele Formatzeilen einfügbar sind, ist die Anzahl der Kontrollbefehle unbegrenzt.

Zu den Kontrollbefehlen gehört auch das »Merge«-Zeichen. Es ist bei der Erstellung von Serienbriefen unverzichtbar, denn immer dort, wo ein »Merge«-Zeichen steht, wird eine Information aus der Arbeitsseite (work page) eingefügt. Dort können, entweder von Hand eingegeben oder aus einer Datenbank Adressen eingeladen werden. Die Position »Global/Fill« im Druckermenü (Bild 3) sorgt dann dafür, daß der Brief entsprechend oft mit den eingefügten Daten ausgedruckt wird. Selbstverständlich können auch beliebig viele Dokumente aneinandergehängt werden. Die einzige Grenze, die dem pausenlosen Drucken gesetzt ist, ist die Speicherkapazität der Diskette.

Die wichtigste aller Funktionen eines Textverarbeitungsprogramms ist aber das Ausdrucken. Und hier trumpft »Vizawrite 64« erst richtig auf.

Eingebaute Centronics-Schnittstelle

Vizawrite 64 hat ein eigenes, eingebautes Centronics-Soft-Interface, das den Anschluß eines Druckers mit dieser Schnittstelle problemlos macht. Die einzigen zusätzlich entstehenden Kosten bestehen in einem 50-Mark-Kabel. Für den Anschluß eines CBM-Druckers wird dieses Kabel natürlich nicht benötigt. Leider kann Vizawrite 64 auf diesen Druckern noch keine deutschen Umlaute erzeugen. Dafür arbeiten aber fast alle anderen Drucker und Schreibmaschinen mit Vizawrite ausgezeichnet zusammen. Dabei ist es fast ohne Belang, mit welchem Interface der Drucker angeschlossen ist. Der Centronics-Interface-Vergleich in Heft 7/84 zeigte, daß Vizawrite 64 als einziges Textverarbeitungsprogramm mit allen Schnittstellen einwandfrei funktionierte. Auch die Original-Epson-Schnittstelle ist geeignet.

Ganz besonders positiv sind die Handhabung und die Hilfen während des Ausdrucks. Zum einen kann der Ausdruck jederzeit gestoppt, fortgesetzt oder neu begonnen werden. Zum anderen helfen sinnvolle Hinweise beim Ausdrucken mit Einzelblatteinzug. Dabei steht vor dem Ausdrucken noch zur Wahl, ob in Blocksatz gedruckt oder welcher Zeilen- beziehungsweise Zeichenabstand gelten soll. Papierlänge, linker Rand, automatischer Seitenvorschub und Anfügen einer Kopfzeile auf jeder Seite sind weitere Funktionen, die einfach und schnell im Druckermenü bestimmbar sind.

Vizawrite 64 ist enorm zuverlässig. Außer mit brutaler Gewalt (Resetknopf) ist das Programm kaum zum Abstürzen zu bringen. Ein einziges Mal während des Tests dachten wir, Vizawrite hätte sich »aufgehängt«.

Dies war der Fall, als die »Shift«- und die »CBM«-Taste zu gleicher Zeit gedrückt wurden. Es war danach nur noch wirres Zeug auf dem Bildschirm. Ein erneuter Druck auf die gleichen Tasten stellte aber den Originalzustand wieder her (eine hervorragende Möglichkeit, Texte blitzschnell vor fremden Blicken zu schützen).

Bei so vielen positiven Eigenschaften sind auch ein paar negative Eigenheiten von Vizawrite 64 anzumerken: Es wurde keine vollständige DIN-Tastatur implementiert »Blindschreiber« müssen sich also etwas umgewöhnen. Die deutschen Umlaute gibt es nur auf Druckern von Fremdherstellern. Und letztlich ist es schade, daß ein so hervorragendes Produkt noch in englisch geschrieben ist. Ein Umstand, den der Hersteller möglichst bald ändern sollte.

Mit 320 Mark für die Diskettenversion und 360 Mark für die Modulversion ist Vizawrite 64 sicherlich nicht billig. Gemessen an den Leistungen ist dieser Preis aber mit Sicherheit mehr als gerechtfertigt.

Auch für Unternehmer interessant

Vizawrite 64 ist nicht nur für den ambitionierten Privatanwender interessant, sondern auch für all jene Unternehmen, die mit möglichst wenig Investitionskosten ein leistungsfähiges Computersystem erwerben wollen. Der Commodore 64 zusammen mit Diskettenlaufwerk, einem Monitor und einem Epson FX 80 kostet zum Beispiel inklusive Vizawrite 64 weniger als 4000 Mark. Den Löwenanteil mit über 1500 Mark trägt dabei der Drucker. Für einen PC mit gleichem Drucker (und der bestimmt ja schließlich das Druckbild) ist mindestens das Doppelte bis Dreifache zu bezahlen. Das Endprodukt, der ausgedruckte Text, unterscheidet sich aber in keiner Weise von dem mit einem PC erstellten. Auch steht die Geschwindigkeit und der Komfort bei der Eingabe einem PC kaum nach.

In der nächsten Ausgabe werden wir ein weiteres Produkt dieser Reihe vorstellen: Vizastar, eine kombinierte Datenbank und Tabellenkalkulation. Der Commodore 64 ist ein preiswerter, aber vielseitiger PC — Sie dürfen gespannt sein.

(Arnd Wängler/aa)

Bezugsquelle: Microton, Postfach 40, CH-2542 Pieterlen

Lohnsteuerjahresausgleich leichtgemacht

Der Kampf mit den Formularen für den Lohnsteuerjahresausgleich ist mühsam und nervenaufreibend. Mit dem Programm HL-LOJA-64 soll das anders werden.

Die Autoren dieses Programms haben sich an ein heikles Thema herangewagt. Die mitgelieferte Beschreibung ist recht spärlich ausgefallen. Doch diesen Nachteil fängt das Programm mit einer einfachen Bedienbarkeit, leicht verständlichen Texten und einer sich stets wiederholenden Korrekturabfrage ab. Als Muster sind dem Handbuch eine Lohnsteuerkarte und ein Formular für den Lohnsteuerjahresausgleich (für das Jahr '83) beigelegt.

Das Programm

Nach dem Starten des Programms werden zuerst die persönlichen Daten, wie Geburtsdatum, Anzahl der Kinder und die Kirchenzugehörigkeit abgefragt. Dann werden die Eintragungen, die sich auf der Steuerkarte befinden, in das Programm übertragen. Auch hier kann man ständig Fehleingaben korrigieren.

Nachdem dieser Programmteil durchlaufen ist, wird mit Hilfe des nächsten Teiles die Anlage N des Formulares ausgefüllt. Das Programm fragt einzeln die steuerrelevanten Daten ab, und weist darauf hin, welche Einträge vorzunehmen sind.

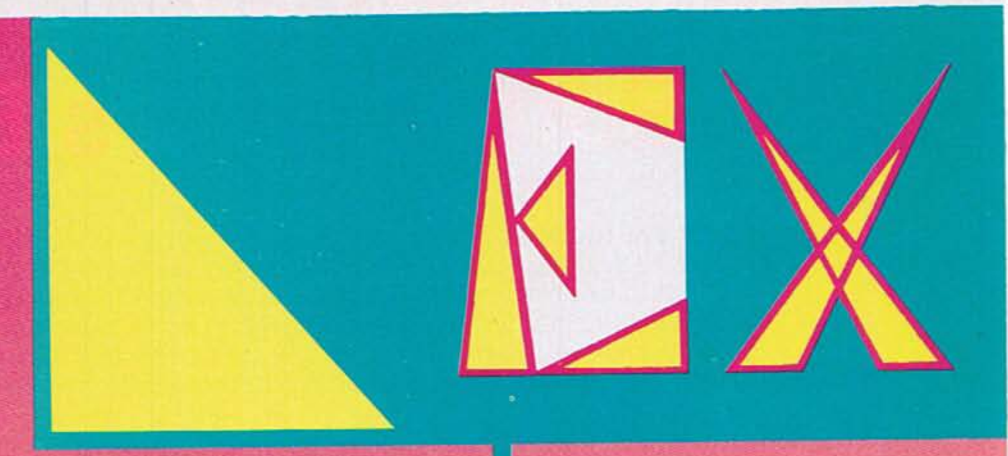
Ist die Anlage N ganz ausgefüllt, wird der Hauptvordruck in Angriff genommen. Das Verfahren beim Ausfüllen ist das gleiche wie bei der Anlage N.

Nachdem das Ausfüllen mit der Unterschrift unter den Antrag abgeschlossen ist, berechnet das Programm die fälligen Steuern und den Betrag, den man nachzahlen muß oder erstattet bekommt.

Fazit

Über die Zweckmäßigkeit dieses Programms kann man geteilter Meinung sein. Solange es nur als Hilfestellung zum Ausfüllen des Jahresausgleichs benutzt wird, ist der Preis von 98 Mark sicherlich nicht zu hoch (man kann auch diesen Kaufpreis unter »Steuerberatungskosten« absetzen). Erwartet man allerdings ein Programm, das den eventuell nötigen Gang zum Steuerberater ersetzt, wird man enttäuscht. Weder Handbuch noch Programm geben konkrete Hinweise über Absetzbarkeit mancher Kosten. Man sollte also möglichst genau wissen, was und wieviel man in dem Antrag an Kosten ansetzen muß. In diesen Fällen wird das Programm wirklich voll ausgenutzt. Allerdings können die Berliner damit recht wenig anfangen. Die Berlinvergünstigung ist nicht berücksichtigt. Vor Änderungen der Steuergesetzgebung braucht man keine Angst zu haben. Das Programm wird, wie der Hersteller mitteilt, bei Änderungen aktualisiert und bei Einsendung einer Originaldiskette ausgetauscht. (rg)

Bezugsquelle: HL Computer-Software GmbH, Hammstr. 8, 6842 Bürstadt, Tel. 06206/8198



Jeder Programmierer möchte mit wachsender Erfahrung seinem Floppy-Laufwerk und seinen Disketten auch die letzten Geheimnisse entreißen. Mit Ex-DOS & Disk Doctor erhält er ein sehr leistungsfähiges Werkzeug dazu.

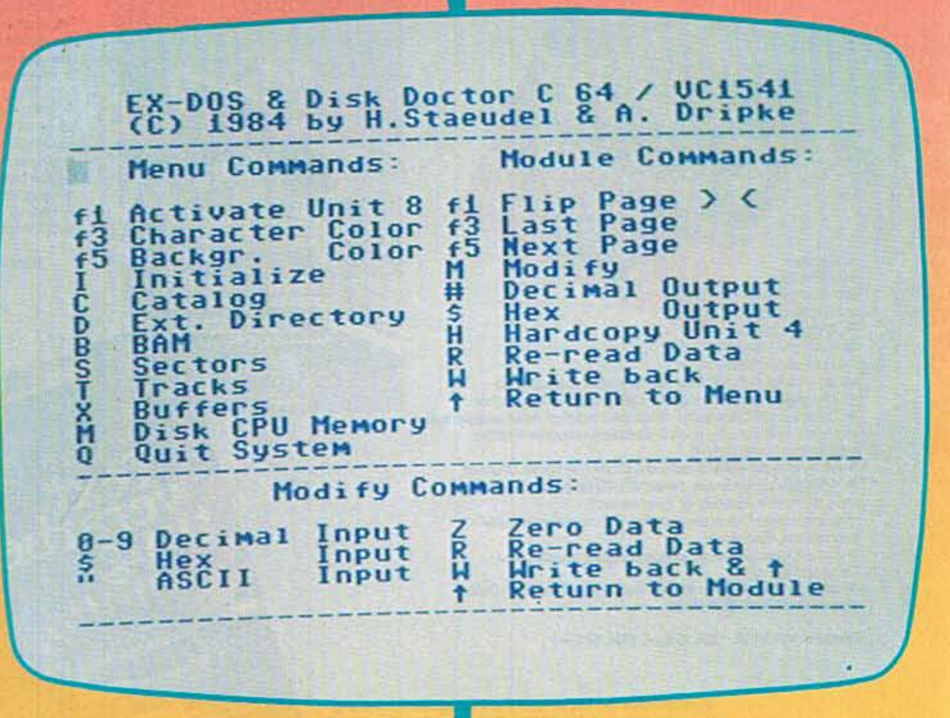
Bestellt man von Interface Age das Programm Ex-DOS & Disk Doctor (138 Mark), erhält man ein gutes, zirka 50 Seiten umfassendes deutsches Handbuch in Heftform, sowie eine gut verpackte Diskette. Gut verpackt heißt, daß in der Diskettenhülle eine Metallscheibe (5 ¼-Zoll-Größe) direkt unter der Diskette liegt. So wird auf einfache Weise eine Beschädigung beim Versand verhindert. Genausogut wie die Verpackung ist auch das Programm. Allerdings muß dazu gesagt werden, daß dieses Programm nicht für Anfänger geeignet ist, die noch keine Ahnung vom Arbeiten mit der Floppystation haben. Vielmehr wird der Inhalt des Commodore-Floppy-Handbuchs zum Verständnis des Ex-DOS & Disk Doctors als bekannt vorausgesetzt. Hat man jedoch diese Kenntnisse, so kann man sie mit Hilfe von Ex-DOS & Disk Doctor voll ausnutzen.

Was das Programm kann

Das Programm ist im großen und ganzen menügesteuert. Jede der vom Menü aus ansteuerbaren Möglichkeiten wird vom Hersteller Programm-Modul genannt. Ex-DOS besitzt folgende Module:

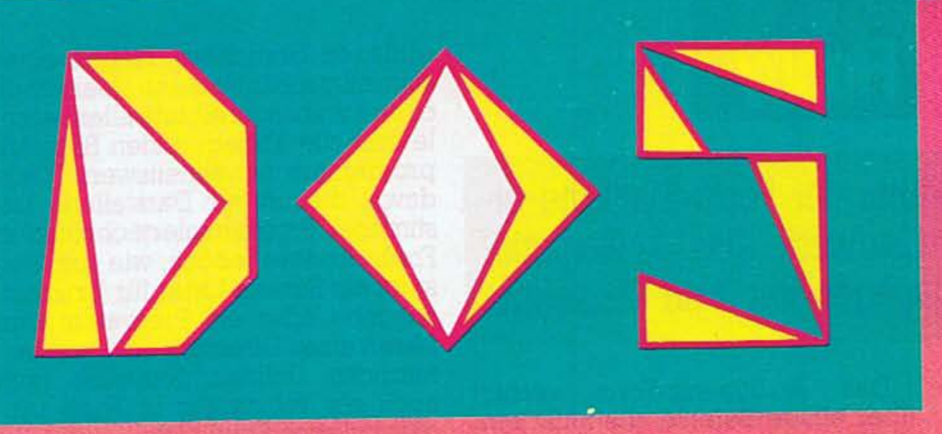
- C Catalog
- D Extended Directory
- B BAM
- S Sectors
- T Tracks
- X Buffers
- M Disc-CPU-Memory

Das Catalog-Modul listet das Inhaltsverzeichnis der Diskette auf, wobei die Ausgabe mit der SPACE-Taste unterbrochen werden kann. Das »Extended Directory« bietet hier noch mehr: Man kann im Inhaltsverzeichnis alles, was das Herz begehrt, ansehen und beliebig ändern, und beispielsweise ohne Schwierigkeiten den Namen eines Files, den Filetyp (PRG, SEQ, USR, DEL, REL), die angezeigte Blockzahl sowie den Anfangstrack und -sektor eines Files manipulieren. Auf einen einfachen Tastendruck hin »locked« man Files, das heißt man schützt sie vor dem Löschen und Umbenennen. Ebenso einfach läßt sich der ganze Vorgang auch wieder umkehren. Sollten Sie Lust haben, die Programmnamen nach Directory-Reihenfolge durchzunumerieren, sie mit Leerstellen aufzufüllen oder zu sortieren, dann genügt ein Tastendruck. Genauso einfach entfernt man Lücken zwischen den Einträgen oder fügt Platz für weitere Namen im Directory ein. Beim Sortieren der Files ist es allerdings lästig, daß die REL-Files im Inhaltsverzeichnis an den Anfang gesetzt werden. Denn: Will ich mit LOAD »*«, 8 das erste Programm von einer Diskette laden, wird ein FILE NOT FOUND ERROR ausgegeben, weil das erste vorhandene File kein PRG-File ist. Hier wäre eine bessere Reihenfolge von Vorteil.



Mit diesem Bild meldet sich Ex-DOS. Es zeigt zu den anwählbaren Modulen auch eine Erklärung der wichtigen Kommandos.

Das nächste Programm-Modul, der Zugriff auf die Block-Availability-Map, bietet folgende Möglichkeiten: Man kann nach freien und belegten Sektoren suchen, einzelne Blöcke belegen und wieder freimachen. Mit der Option »Extended BAM« läßt sich außerdem nach Blöcken su-



Der Disketten — Doktor

chen, die zwar Daten enthalten, jedoch nicht belegt sind. Diese Funktion hilft, Daten von Disketten zu retten, bevor sie versehentlich überschrieben werden.

Ganz radikalen »Disk-Murksern« gelingt es mit einer zusätzlichen Option, eine Disk durch das Freigeben aller Sektoren in der BAM völlig zu verunstalten.

Eines der wichtigsten Unterprogramme ist das Sectors-Modul. Hiermit kann man seine Disks endgültig ruinieren — oder auch reparieren. Der Inhalt des angewählten Blocks wird sowohl in Hex (per Tastendruck auch Umschaltung auf Dezimal möglich) als auch in ASCII-Darstellung (Text) angezeigt. Innerhalb dieses Blocks kann man beliebig mit dem Cursor herumfahren und die Werte ändern, sowie auf der Diskette vorwärts und rückwärts blättern. Dabei kann sowohl nach logischer als auch nach pysikalischer Reihenfolge vorgegangen werden. Logisch bedeutet, daß nicht nach der Durchnumerierung der Tracks und Sektoren geblättert wird, sondern der Computer nach der vorgegebenen Reihenfolge Track/Sektor Zeigerkette vorgeht. Das Suchen nach dem nächsten belegten Block sowie das Kopieren eines einzelnen Sektors ist ebenfalls möglich (jetzt können sich die Raubkopierer sogar noch an den bisher verschont gebliebenen Sektoren vergreifen).

Das Tracks-Modul bietet im großen und ganzen die gleichen Funktionen wie das Sectors-Modul. Auf dem Bildschirm werden immer die ersten sieben Bytes eines jeden Sektors eines Tracks angezeigt. In ihnen sind wichtige Informationen enthalten, wie zum Beispiel Startadressen, Blockverbindung, und so weiter. Insbesondere diese Tatsache hilft sehr beim Reparieren von Disketten, die aus Versehen gelöscht wurden (zum Beispiel durch Formatieren ohne ID).

Die Module »Buffers« und »Disk-CPU-Memory« erlauben den direkten Zugriff auf jedes einzelne Byte des Disk-Controllers. Sie sind nur mit entsprechenden Kenntnissen oder Unterlagen (wie beispielsweise DOS-Listing) sinnvoll einzusetzen. Die falsche Benutzung kann zu einem Absturz der Floppy führen. Außerdem sollte darauf geachtet werden, daß man nicht den im Floppy-RAM vorhandenen Teil des Ex-DOS & Disk Doctors überschreibt.

Die Dokumentation

Doch nun genug aufgezählt, die Funktionen des Programms sind ja schließlich auch in einem sehr ausführlichen Handbuch erläutert, das nicht nur genau auf alle Fähigkeiten des Disk Doctors eingeht, sondern auch möglicherweise auftretende Fehler behandelt. Als ideale Mischung könnte man das Data-Becker-Floppybuch mit Ex-DOS & Disk Doctor bezeichnen.

Zusätzlich zu den Befehlserläuterungen gibt es noch einige Anhänge im Handbuch, von denen der interessanteste wohl der Leitfaden zur Diskettenreparatur ist. Es handelt sich hier um eine genaue Schritt-für-Schritt-Anleitung, die auch dem weniger versierten Programmierer eine Diskettenreparatur ermöglicht. Nach dieser Anleitung folgen einige Referenzseiten über Diskettenaufbau, Blockformate, sowie nähere Angaben zu den CBM 8050-, 8250-, 4040/4031-Floppylaufwerken — aber keine Angaben zum VC 1541!

Trotz aller Informationen, Erklärungen von Fachausdrücken etc. wird der unerfahrene Floppy-Anfänger also erst einmal stutzig. Das größte Manko am Ex-DOS & Disk Doctor besteht darin, daß die Anleitung hauptsächlich noch für die alten CBM-Computer geschrieben ist. Eine kleine Anpassung für die C 64-Version in Form eines kleinen Zusatzheftchens ist durchaus angebracht. Schließlich wird auch für die C 64-Version des Exbasic Level II ein Zusatzhandbuch mitgeliefert.

Dem routinierten Programmierer wird es jedoch kaum schwerfallen, das für die VC 1541 richtig umzusetzen. Es ist offensichtlich, daß der C 64 den Befehl DLOAD (Basic 4.0) nicht besitzt, und die Basic-Startadresse nicht durch die Werte 1 und 4, sondern durch 1 und 8 auf der Disk angegeben ist. Welche Angaben nur für die alten CBMs gelten, wird man schnell herausfinden.

Eine empfehlenswerte Sache

Zusammen mit den beiden mitgelieferten Hilfsprogrammen 1541-Backup sowie der Anpassung der Bildschirm-Hardcopy-Option auf den Printer/Plotter 1520 ist Ex-DOS & Disk Doctor als das beste Disk-Utilitie zu bezeichnen, das derzeit im Fachhandel erhältlich ist. Den kleinen Fehler im Handbuch wird Interface Age sicher bald verbessern.

(M. Kohlen/gk)

Info: Interface Age, Vohburger Str. 1, 8000 München 21

Forth ohne Floppy

Die virtuelle Speicherverwaltung der Forth-Systeme erfordert meistens ein Floppy-Laufwerk. Das Audiogenic-Forth benötigt dagegen nur eine Datasette.

Der Befehlsumfang des Audiogenic-Forth beschränkt sich im wesentlichen auf den FIG-Forth-Standard und ist in einem einzigen Vokabular zusammengefaßt. Dies bedeutet im Vergleich mit anderen Forth-Versionen eine gewisse Einschränkung. Mancher Benutzer wird spezielle Befehle zum Ansprechen eines Floppy-Laufwerkes vermissen. Dieses Manko erklärt sich aus der Zielgruppe, an die sich diese Forth-Version primär wendet: An jene Computerbesitzer, die ohne Floppy-Laufwerk auskommen müssen und sich lediglich einer Datasette bedienen. Dementsprechend wurden die Befehle für die Floppy einfach fortgelassen.

Doch wenden wir uns der Speicherverwaltung zu. Im Gegensatz zur sonst üblichen virtuellen Speicherverwaltung wurde bei dieser Forth-Version eine »screen compressing routine« implementiert. Diese Routine reduziert zum Beispiel einen leeren Screen auf eine Länge von nur 5 Bytes. So hat man alle 255 möglichen Screens ständig im Speicher, also im direkten Zugriff. Ein ständiges Laden und Speichern von und auf Diskette ist somit überflüssig.

Das Textfeld Nummer Null fungiert als Interpreterscreen. Hier werden alle Benutzereingaben sofort ausgewertet und gegebenenfalls compiliert. Beim interaktiven Arbeiten mit dem System stört es häufig, daß alle auf dem Bildschirm ausgegebenen Ergebnisse nach Betätigen einer Taste (für die nächste Eingabe) wieder verschwinden.

Ebenso wie Screen 0 werden auch die übrigen Textfelder mit einem zeilenorientierten Editor bearbeitet, der 16 Zeilen mit je 64 Spalten durch horizontales Scrollen auf der Mattscheibe darstellt. Dieser Editor kann zwar einen Full-Screen-Editor nicht ersetzen, aber aufgrund mehrerer Hilfsfunktionen und Editiermöglichkeiten, die zum Teil auch auf die Funktionstasten gelegt sind, wird ein insgesamt annehmbarer Bedienkomfort erreicht. Die Funktionstaste F1 ist übrigens frei belegbar.

Das Audiogenic-Forth verfügt über einige Befehle, die nicht zum FIG-Forth-Standard gehören. Sie unterstützen hauptsächlich die Hardwaremöglichkeiten des VC 20 beziehungsweise des C 64. Die VC 20-Version bietet dabei sogar etwas mehr, denn hier wird die Programmierung von Grafik und Musik durch die Befehle SOUND, VOLUME und COLOUR vereinfacht. Besondere Beachtung verdient die Behandlung der RESTORE-Taste. Durch den Befehl »RESTORE« wird wahlweise keine Reaktion, eine Rückkehr in den Interpreterscreen oder ein Kaltstart ausgelöst.

Das Standard-Vokabular von Audiogenic-Forth

Stackmanipulation
OVER DROP SWAP DUP ROT ?DUP SP@ SP! RP! >R R> R@ I J K STACK-BOTTOM

Numerische Befehle
DIGIT + D+ MINUS DMINUS MIN MAX AND OR XOR U* U/ 1+ 2+ 1- 2- - HEX DECIMAL LITERAL NUMBER S- ->D +- D+- ABS DABS * 2* 2/ MOD /MOD M/MOD / M**/ */MOD 0 1 2 3 -1 DPL BASE 0= 0< = U< < > HOLD <# #> SIGN # #S

Textfeld-Speicherung
TP SCR BLK EXP COM EXPAND SSAVE SLOAD DEVICE ?SCREENS DELETE COPY RESET CLR0 - -> (LOAD SCREEN SCRSTART C/L CONTROL

Speicherzugriff
FILL ERASE BLANKS CMOVE <CMOVE @ C@ ! C! TOGGLE +! , C, COUNT

Wörterbuch
FENCE SET-FORTH -FIND HERE ALLOT LATEST TRAVERSE LFA CFA NFA PFA FORGET ' + ORIGIN PAD VOCABULARY DEFINITIONS DP CURRENT CONTEXT

Compiler-Anweisungen
>IN EXECUTE ENCLOSE ERROR ?ERROR ?STACK ?COMP ?EXEC ?PAIRS ?CSP ?LOADING COMPILE [COMPILE] [ ] WORD !CSP ;S QUIT ABORT INTERPRET IMMEDIATE STATE SMUDGE COLD LITERAL

Ein-/Ausgabe
KEY 0KEY EMIT CR TYPE -TRAILING (.») .» EXPECT QUERRY SPACE SPACES D.R D. .R U. . ? CLR ?TERMINAL TEXT #IN P HOME BL TIB OPEN CLOSE IN OUT OFF STATUS

Definitions-Worte
: ; CREATE CONSTANT VARIABLE <BUILDS DOES>

Kontrollstrukturen
BRANCH 0BRANCH (LOOP) (+LOOP) (DO) LEAVE DO LOOP +LOOP BACK BEGIN UNTIL AGAIN IF ELSE THEN REPEAT WHILE DOBRAN I J K R@

VC 20 / C 64 - Funktionen
SOUND VOLUME NMI COLOUR SET RESTORE

Das Handbuch, das in Umfang und Ausführlichkeit stark an die Commodore-Handbücher erinnert, ist leider in Englisch geschrieben und kann ein Forth-Lehrbuch nicht ersetzen. Es ist allerdings sehr gut gegliedert und zum schnellen Nachschlagen brauchbar. Neben einer sinnvoll geordneten und übersichtlichen Kurzbeschreibung aller Befehle sind die abgedruckten Beispielprogramme erwähnenswert. Viele davon dienen zur Darstellung bestimmter Programmiertechniken in Forth. Andere jedoch, wie zum Beispiel ein Screen-Lister für Druckerausgabe oder ein Programm zum Lesen einer Directory sind wirklich nützliche Utilities. Natürlich fehlt auch ein vollständig in Forth programmiertes Spiel nicht.

Das Audiogenic-Forth gibt es als Steckmodul in einer Ausführung für den C 64 und gleich in zwei Versionen für den VC 20, einmal mit und einmal ohne eingebaute 3 KByte Speichererweiterung. Die VC 20-Version ohne eingebaute Speichererweiterung ist jedoch auf der Grundversion nicht lauffähig und kann nur mit einem bereits erweiterten VC 20 benutzt werden.

Empfehlenswert ist das Audiogenic-Forth vor allem für Systeme ohne Floppy-Laufwerk, die bisher die mächtige Sprache Forth aufgrund der nötigen Zwischenspeicherung der Screens auf Diskette nicht benutzen konnten.

(Daniel Kobler/ev)

Audiogenic-Forth ist erhältlich über Kingsoft, Fritz Schäfer, Schnackebusch 4, 5106 Roetgen.
Preis 119 Mark (Steckmodul).



So macht man Basic-Programm

Basic-Programme können nach drei Gesichtspunkten optimiert werden: strukturiert und lesbar, schnell sowie speicherplatzsparend. Nach der Serie über strukturiertes Programmieren wird in diesem Beitrag der Aspekt der Laufgeschwindigkeit und ihrer Verbesserung behandelt. Alle Aussagen und Beispiele gelten sowohl für den VC 20 als auch den C 64.

Laufzeiten der Programmversionen für »Buchstaben auf dem Bildschirm«

Version Nr.	Programmier-Methode	Laufzeit (Sek.) VC 20	C 64
1	Buchstaben POKEn, Zählschleife mit IF ... THEN	8,25	10,48
2	POKE-Adressen vordefinieren	5,78	7,30
3	alle Variablen vordefinieren	5,15	6,15
4	Variablen mit 2 Buchstaben	4,71	5,63
5	Variable mit 1 Buchstaben	4,63	5,51
6	PRINT CHR $ statt POKE	3,40	4,05
7	CHR $ vordefinieren	2,58	3,10
8	PRINT "A"	2,53	3,03
9	Schleife mit IF-GOTO statt IF-THEN	2,51	3,00
10	IF Z< >S statt IF Z = S	2,50	2,98
11	Schleife mit FOR-NEXT	0,91	1,08
12	NEXT ohne Zählvariable	0,83	1,00
13	Listing mit REMs	0,98	1,20
14	Listings ohne REM, ohne Abstände	0,81	0,98
15	Alles in einer Zeile	0,78	0,93
16	Maschinensprache	0,033	0,066

Es gibt einige Dinge auf der Welt, die man sehr wohl einzeln, aber nicht alle gleichzeitig haben kann. Wirtschaftswachstum, keine Inflation und niedrige Arbeitslosigkeit lassen sich eben nicht unter einen Hut bringen.

Bei der Computerei, oder genauer gesagt beim Prorammieren, gibt es in einer höheren Sprache wie Basic ebenfalls so ein magisches Dreieck: strukturierte Programme, minimaler Speicherbedarf, kürzere Laufzeiten.

Gut lesbare und klar gegliederte Programme brauchen oft mehr Speicherplatz als es sich zum Beispiel Besitzer der VC 20-Grundversion leisten können. Deswegen soll mein heutiger Beitrag Sie zum Expe-



rimentieren anregen, mit welchen verschiedenen Methoden Basic-Programme schneller gemacht werden können.

Basic ist nicht immer gleich Basic

Die Commodore-Handbücher sagen leider zu diesem Thema recht wenig. Ich bitte Sie an Ihrem Computer, VC 20 oder C 64, Platz zu nehmen.

Als erstes wollen wir uns einen einfachen aber typischen Programmablauf überlegen, welchen wir mit mehreren Basic-Möglichkeiten programmieren können. Übrigens: Wegen der guten Lesbarkeit schreiben Sie die nachfolgenden Basic-Zeilen mit Leerstellen zwischen den Befehlen.

Ich weiß, es geht auch kürzer, aber bei meinen Experimenten spielt Speicherplatz keine Rolle und außerdem habe ich noch einen Hintergedanken, den ich erst später erklären kann. Nicht zuletzt will ich dadurch auch erreichen, daß die Laufzeiten der Programme mit den Ihren übereinstimmen.

Schalten Sie bitte auch Programmierhilfen (Toolkit, Simon's Basic etc.) und Disk Operating-Programme (DOS 5.1) aus, denn sie verlangsamen den Programmablauf.

Die interne Uhr mißt die Zeit

Beide Computer, VC 20 und C 64, haben eine interne Uhr, deren Zeit abgefragt, ausgedruckt und somit zur Zeitmessung verwendet werden kann. Im Befehlssatz der Commodore-Handbücher finden Sie dazu die beiden Funktionen TI und TI$. Mein Zeitmessungsprogramm besteht aus zwei Zeilen. Die »Stoppuhr« wird gestartet mit:

```
10 TI$="000000"
```

Sie wird am Ende des Testprogramms gestoppt mit:

```
1000 PRINT TI/60 "SEKUNDEN"
:END
```

Zwischen diese beiden Zeilen stecken wir dann die Prüflinge, das heißt die Programme, deren Laufzeit wir messen wollen.

Zur Prüfung geben wir zuerst die Zeile 15 mit einer simplen Verzögerungsschleife ein.

```
15 FOR T=1 TO 100:NEXT T
```

Das Programm dieser drei Zeilen hat die Laufzeit von 0,13 Sekunden mit dem VC 20 und 0,15 Sekunden mit dem C 64.

Das erste Testprogramm

Löschen Sie bitte wieder die Zeile 15. Als Programm, welches wir in mehreren Versionen programmieren wollen, habe ich mir einen Ablauf ausgesucht, der auch optisch verfolgbar ist. Auf dem Bildschirm soll nämlich der Buchstabe A gleich 374 mal nebeneinander gedruckt werden.

Die Zahl 374 hat nichts Magisches an sich. Es sind ganz einfach 17 Zeilen voller A's auf dem Schirm des VC 20, der mit seiner begrenzten Spaltenzahl hier den Ton angibt. 17 Zeilen lassen uns genug Platz, um die gestoppte Zeit darunter gut lesbar anzuzeigen.

Ich bleibe für den C 64 bei derselben Zahl, damit wir beide Computer miteinander vergleichen können und damit die Programme möglichst identisch sind.

In der **Version 1** des Programms verwenden wir POKE-Befehle, mit denen wir das A und die dazugehörige Farbe auf den Bildschirm, das heißt in den Bildschirmspeicher (Video-RAM) und den Farbspeicher (Color-RAM) bringen. Beim VC 20 (ohne Erweiterung) beginnt der Bildschirmspeicher ab Speicherzelle 7680, der Farbspeicher ab 38400. Beim C 64 sind es die Speicherzellen 1024 und 55296. Schauen Sie bitte in den Commodore-Handbüchern nach und vergewissern Sie sich, daß Sie dieses System des Zeichen-POKEns verstehen. Es ist dort gut beschrieben. Der Buchstabe A hat den Bildschirm-Codewert 1. Als Farbe wähle ich die »Normalfarben« der beiden Computer. Der Farbcode für das Blau des VC 20 ist 6, für das Hellblau des C 64 ist er 14.

Das Programm beginnt mit dem Löschen des Bildschirms (Zeile 20) und POKEt dann in Zeile 40 das erste A in den ersten Platz des Bildschirms.

```
20 PRINT CHR$ (147)
40 POKE 7680+Z:POKE 38400+Z,6
(40 POKE 1024+Z:POKE 55296+
Z,14)
```

Die Zeile in Klammern gilt für den C 64. In Zeile 40 finden Sie zusätzlich eine Variable »Z«. Wie geplant, soll das A 374mal gePOKEt werden. Also müssen wie die Zahl der Speicherzelle laufend um 1 erhöhen. Dazu erfinden wir diese Variable Z, die zur Speicherzelle addiert wird.

Wir setzen Z am Anfang des Programms (in Zeile 30) auf Null und zählen es in Zeile 50 um 1 weiter.

```
30 Z=0
50 Z=Z+1
```

Als nächstes müssen wir prüfen, ob Z den Schlußwert 374 erreicht hat. Wenn nicht, dann soll der nächste POKE-Befehl ausgeführt werden, das heißt wir springen auf Zeile 40 zurück. Dann kommt die Zeile 1000 zum Zug mit dem Stoppen und Ausdrucken der Laufzeit. Also:

```
60 IF Z=374 THEN 1000
70 GOTO 40
```

Ich schlage vor, daß Sie das kleine Programm nochmal LISTen, damit wir es komplett sehen können.

Die Laufzeit wird nur davon beeinflußt, was zwischen den Zeilen 10 und 1000 steht. Sie können vor der Zeile 10 dem Programm hinzufügen, was Sie wollen.

Ein erster Probelauf mit RUN bringt das gewünschte Ergebnis, nur eins ist noch unschön: Der PRINT-Befehl in Zeile 1000 druckt uns die Zeit oben in die 2. Zeile, wo sie schlecht erkennbar ist. Wir könnten sie in einer anderen Farbe drucken, aber ich habe einen besseren Vorschlag.

Ersatz für den Befehl »PRINT-AT«

Wir brauchen ein Kochrezept, um mit einem PRINT-Befehl an einen ganz bestimmten Platz auf dem Bildschirm drucken zu können. Einige Basic-Dialekte kennen den Befehl »PRINT-AT«. Welche Möglichkeiten bietet uns das Basic von Commodore?

1) PRINT"[Cursor Down] [Cursor Right]"
2) PRINT TAB(X)
3) PRINT SPC(X)

Um zum Beispiel an den 3. Platz in der 20. Zeile die Zeit zu drucken, müßten wir 18mal das inverse Q für Cursor Down und 1mal Cursor Right eingeben.

Mit TAB (X) geht es besser. Wir haben nur ein Problem, daß nämlich der höchste zulässige Wert für X nur 255 ist (X nennt man das »Argument«). Wir müssen deshalb zwei TAB-Befehle hintereinander setzen, um einen Abstand von 400 Leerstellen zu erzeugen.

```
1000 PRINT TAB (255) TAB (155) TI/60"SEKUNDEN":END
```

Für das Argument von SPC gilt dieselbe Begrenzung von 255. Eine doppelte Verwendung von SPC geht natürlich auch, allerdings zählt SPC nicht vom Anfang der Zeile, sondern ab der letzten Cursor-Stelle. Durch Rechnen oder einfach durch Probieren finden wir die Gesamtzahl von 397, das gibt:

```
1000 PRINT SPC (255) SPC (142) TI/60"SEKUNDEN":END
```

Es gibt noch eine dritte Methode, um PRINT-AT zu simulieren.

In die Speicherzelle 214 kann die Zahl einer Zeile hineingePOKEt werden, auf die mit einem nachfolgenden PRINT der Cursor gesetzt wird. Das gleiche gilt für einen Platz in einer Zeile mit der Speicherzelle 211. Versuchen Sie es mit der direkten Eingabe:

```
POKE 214,8:PRINT:POKE 211,4:PRINT"A"
```

Drei Formen für »PRINT AT«

Das druckt den Buchstaben A in die 4. Spalte auf der 9. Zeile. Für unseren Anwendungsfall in Zeile 1000 müssen wir die Zahl 18 nach 214 POKEn, 211 können wir vernachlässigen.

```
1000 POKE 214,18:PRINT:PRINT TI/60 "SEKUNDEN":END
```

Alle drei Methoden sind gleichwertig, sowohl in Auswirkung als auch beim Speicherbedarf. Ich bleibe im folgenden bei der 214-Methode.

Zurück zur Version 1 des Testprogramms. Das Programm unterscheidet sich für die beiden Computer nur in der Zeile 40, allerdings auch durch die Laufzeit. Nach RUN erhalten wir mit dem VC 20 8,25 Sekunden, mit dem C 64 10,48 Sekunden. Dieses Auffüllen des Bildschirms mit A geht halt recht langsam. Schon beim Zuschauen wird man ungeduldig. Der Teil in diesem Programm, welcher die Laufzeit am nachhaltigsten beeinflußt, ist die 374fache Wiederholung des POKE-Befehls. Bei einem POKE-Befehl ist die Umwandlung der Zahlen aus dem ASCII-Code sehr zeitaufwendig. Hoppla, was heißt denn das schon wieder, sagen Sie jetzt vielleicht.

Jede Zahl, wie zum Beispiel 7680 oder 1024, wird zuerst als vier einzelne ASCII-Codezahlen gespeichert. Wenn das Programm abläuft, werden diese ASCII-Zahlen zuerst in ganze Zahlen, dann in Fließkomma-Zahlen umgewandelt — das für den Fall, daß mit den Zahlen arithmetische Funktionen ausgeführt werden. Schließlich werden sie wieder in eine ganzzahlige POKE-Adresse umgewandelt, und das in unserem Fall 374mal!

Hier können wir einen ersten innerbetrieblichen Verbesserungsvorschlag einreichen. Wenn wir eine so häufig vorkommende Zahl wie die POKE-Adresse in Zeile 40 am Anfang des Programms einer Variablen zuweisen, dann erfolgt die oben genannte Umwandlungssequenz nur einmal, nämlich am Anfang des Programms. Das Programm muß dann 374 mal nur den Wert der Variablen im Speicher suchen und das geht viel schneller. Wollen Sie es sehen? Ändern Sie bitte für diese **Version 2** die Zeilen 30 und 40.

Die Anfangsadresse im Bildschirmspeicher definieren wir als Variable mit dem schönen und zutreffenden Namen »VIDEO«, die des Farbspeichers mit »FARBE«. In Zeile 30 erhalten sie dann die Werte

```
30 Z=0:VIDEO=7680:FARBE=38400:REM VC 20
(30 Z=0:VIDEO=1024:FARBE=55296:REM C 64)
```

Zeile 40 ergibt sich dann eigentlich zwangsläufig:

```
40 POKE VIDEO+Z, 1:POKE FARBE+Z,6
(40 POKE VIDEO+Z,1:POKE FARBE+Z,14)
```

Alles andere bleibt gleich. Tippen Sie RUN ein. Ergebnis: der VC 20 braucht 5,78 Sekunden, der C 64 7,3 Sekunden. Wir haben also eine Verkürzung von zirka drei Sekunden erzielt, das sind 30 Prozent!

Sicher ist Ihnen aufgefallen, daß der VC 20 schon wieder schneller ist als der C 64.

In der Tat können die oft mitleidig behandelten VC 20-Besitzer mächtig stolz sein: Der VC 20 ist immer schneller als der C 64 und auch (fast immer) schneller als die »Neuen« C-116 und Plus/4 (und übrigens einer der schnellsten Heimcomputer überhaupt).

Der VC 20 ist der schnellste

Eine Beschleunigung von 30 Prozent ist gut, aber noch nicht alles, was wir erreichen können. Da die Methode der vordefinierten Variablen so effektiv ist, wollen wir sie auf alle oft verwendeten Zahlen des Programms anwenden. Neben Z, VIDEO und FARBE gibt es noch die 1 für den Buchstaben A und die 6 (14) für die Farbe sowie den Schlußwert 374 der Schleife.

Sie wissen schon, wie das geht. Wir ändern folgende Zeilen:

```
30 Z=0: SCHLUSSWERT=374:VIDEO=7680:FARBE=38400:BUCHSTA=1:DRUCK=6
(30 Z=0: SCHLUSSWERT=374:VIDEO=1024:FARBE=55296:BUCHSTA=1:DRUCK=14)
40 POKE VIDEO+Z, BUCHSTA:POKE FARBE+Z,DRUCK
60 IF Z=SCHLUSSWERT THEN 1000
```

Das ist **Version 3** des Programms, mit Laufzeiten von 5,15 Sekunden (VC 20) beziehungsweise 6,15 Sekunden (C 64). Das ist eine Verbesserung von 3,10 (4,23) Sekunden gegenüber der ersten Version.

Bisher haben wir die Variablen im Sinn einer guten Lesbarkeit mit langen und verständlichen Namen versehen. Aber das kostet natürlich Speicherplatz und auch Geschwindigkeit. Der Grund ist immer derselbe: Bei der Variablen VIDEO sind fünf Zeichen 374 mal zu bearbeiten. Wenn wir sie nur VI nennen, ist das erheblich weniger.

In **Version 4** des Programms reduzieren wir also alle langen Variablennamen auf zwei Zeichen. Wir wollen mal schauen, was das bringt:

```
30 Z=0: SC=374: VI=7680:FA=38400:BU=1:DR=6
(30 Z=0: SC=374: VI=1024:FA=55296:BU=1:DR=14)
40 POKE VI+Z,BU:POKE FA+Z,DR
60 IF Z=SC THEN 1000
```

Ergebnis: VC 20: 4,71 Sekunden
C 64: 5,63 Sekunden

Gegenüber der Version 1 des Programms haben wir schon eine Verbesserung von 43% (46%) erreicht.

Sie wissen sicher, daß der Computer alle Variablennamen immer auf zwei Stellen reduziert, daß aber einstellige Variablen durchaus zu-

gelassen sind. Damit müßte unser Programm eigentlich noch schneller werden. Die folgenden Änderungen für **Version 5** zielen genau darauf:

30 Z=0: S=374: V=7680:F=38400: B=1:D=6
(30 S=374: V=1024: F=55296:B=1: D=16)
40 POKE V+Z,B:POKE F+Z,D
60 IF Z=S THEN 1000

Die erzielte Verbesserung ist meßbar, aber nicht gerade überwältigend. Der VC 20 braucht nun 4,63 und der C 64 5,51 Sekunden. Mit dem bisher Gesagten läßt sich bereits eine erste generelle Regel für schnellere Basic-Programme aufstellen.

Regel 1
* Häufig vorkommende Zahlen, Adressen und Variable, besonders innerhalb von Schleifen, werden am Anfang des Programms vordefiniert.
* Variable, die sich im Lauf des Programms verändern, werden trotzdem vordefiniert, allerdings mit einem unschädlichen Anfangswert (dummy).
* Die Zahl, die am häufigsten vorkommt, wird als erste vordefiniert (damit sie schneller »gefunden« wird).
* Variablennamen sollen möglichst einstellig, aber höchstens zweistellig sein.

Halt! Löschen Sie das Programm noch nicht. Wir liegen noch unter 50 Prozent mit unseren Verbesserungen, und das reicht noch lange nicht. Was können wir am bisherigen Programm noch ändern? Denken Sie mal nach. Wie kann man Buchstaben auf den Bildschirm bringen? Natürlich, mit PRINT statt der POKEs.

In **Version 6** des Programms ersetzen wir die Buchstaben-POKE-Befehle durch eine PRINT-Anweisung, die prinzipiell noch den Vorteil hat, daß keine Farbanweisung nötig ist. Den Buchstaben A wollen wir in dieser Version zunächst einmal mit seiner ASCII-Codezahl 65 angeben, ein Semikolon setzt alle A hintereinander.

40 PRINT CHR$(65);

Für die Schleife brauchen wir nur die beiden Variablen Z und S:

30 Z=0:S=374

Alle anderen Zeilen bleiben unverändert.
Nach RUN sehen wir als Ergebnis:
VC 20: 3,4 Sekunden
C 64: 4,05 Sekunden

Das ergibt eine weitere Verbesserung von 1,2 (1,5) Sekunden. Unser Programmbeispiel wird also durch die Verwendung von PRINT statt POKE stark beschleunigt. Das geht natürlich deswegen besonders gut, weil alle Buchstaben automatisch hintereinander gesetzt werden. Wenn wir jedesmal den Platz mit angeben müßten, wohin gePRINTet werden soll, wäre der Vorteil rasch verspielt.

Die Anweisung PRINT CHR$(65) ist zwar gut lesbar, aber wir haben ja vorher gelernt, daß Vordefinieren von Variablen schneller ist. In **Version7** machen wir das auch mit der PRINT-Variablen.

30 Z=0: S=374: A$=CHR$(65)
40 PRINT A$

Das Resultat ist wieder beeindruckend: 2,58 Sekunden beim VC 20, 3,10 Sekunden beim C 64. Das sind schon 69 Prozent Verbesserung gegenüber der 1. Version.

Aber selbst — oder gerade — jedem Anfänger ist die direkte PRINT-Anweisung mit Gänsefüßen am geläufigsten. Sie braucht auch weniger Speicherplatz und macht sogar ein Vordefinieren unnötig. Wir wollen schauen, ob sie auch schneller ist.

Ändern Sie für **Version 8** die Zeilen 30 und 40:

30 Z=0:S=374
40 PRINT "A";

Erstaunlicherweise bringt das fast gar nichts, beim VC 20 nur 0,05 Sekunden Verbesserung, beim C 64 0,07 Sekunden. Die Erklärung liegt darin, daß beide Darstellungen, CHR$(65) und "A" ASCII-Code-Verwender sind. Damit ist der einzige Unterschied zwischen Version 7 und 8 die Anzahl der Zeichen im Programm.

Fassen wir zusammen:

Regel 2
* In Schleifen mit aneinandergereihten Druckanweisungen ist PRINT viel schneller als POKE.
* Die PRINT-Variablen sollen entweder vordefiniert oder im Gänsefuß-Modus eingesetzt werden.

In der PRINT-Schleife (Zeile 40) und nachfolgender IF-Abfrage (Zeile 60) gibt es noch zwei Feinheiten. Basic erlaubt uns bei bedingten Sprüngen statt IF .. THEN GOTO .. nur IF .. THEN .. zu schreiben und das haben wir bisher auch brav gemacht.

Man kann aber auch IF .. GOTO verwenden. Eigentlich ist nicht zu erwarten, daß zwischen den beiden Schreibarten ein Zeitunterschied besteht. Der Fall ist tatsächlich fast akademisch, wie **Version 9** beweist:

60 IF Z=S GOTO 1000

Laufzeit des VC 20: 2,51 Sekunden, die des C 64: 3,0 Sekunden.

Eine andere Änderung bringt auch nur ganz wenig in der Geschwindigkeit, aber sie spart uns eine ganze Zeile und damit Speicherplatz:

60 IF Z< >S GOTO 40
70 entfällt

Diese **Version 10** gewinnt nur 0,01 (0,02) Sekunden.

Regel 3
* Bei Schleifen mit Sprunganweisungen ist IF .. GOTO schneller als IF..THEN
* Prüfung auf Ungleichheit (< >) bietet Vorteile, wenn sie eine Zeile erspart.

Für die etwas weiter Fortgeschrittenen unter Ihnen gebe ich hier noch eine weitere Anregung, die wir mit unserem Prüfprogramm nicht testen können.

Die Prüfung mit IF..THEN hängt oft von mehr als einer Bedingung ab. Zum Beispiel kann sie so lauten:

100 IF (A=1 AND B=2 AND C=3) THEN 999
110 GOTO 50

Zeile 100 prüft jedesmal, ob alle drei Bedingungen erfüllt sind, erst nach dem THEN wird entschieden, ob das Programm auf Zeile 999 springt oder auf 110 weiterläuft. Nehmen wir an, A ist im 20. Durchlauf erfüllt, B im 50. Durchlauf, C aber erst im 300. Durchlauf. Das Programm muß also 300mal alle drei Bedingungen nachprüfen. Wenn wir die Zeile 100 so schreiben:

100 IF C=3 THEN IF B=2 THEN IF A=1 THEN 999

dann bricht das Programm 300mal die Prüfung nach dem C sofort ab und geht in 110 weiter. B und A werden erst dann zur Prüfung herangezogen, wenn C=3 ist. Es ist wohl klar und einzusehen, daß die zweite Schreibweise schneller ist.

Regel 4
Bei IF..THEN-Prüfungen mit mehreren Bedingungen sollen diese Bedingungen in einzelnen IF..THEN-Prüfungen hintereinander gesetzt werden. Dabei wird die Bedingung an die erste Stelle gesetzt, welche als erste nicht erfüllt ist.

Ich habe mit Absicht bis hierher die 374malige Wiederholung der Schleife mit der Zählvariablen Z hochgezählt und mit IF..THEN beziehungsweise IF..GOTO den Schlußwert abgefragt. Das gab mir die Gelegenheit, die Schnelligkeit dieser Methode ganz auszureizen. Und wir haben auch die ursprüngliche Laufzeit von 8,25 (10,48) Sekunden auf 2,50 (2,88) Sekunden reduziert!

Jetzt wollen wir noch eine andere Basic-Möglichkeit austesten, die Sie ja alle kennen, nämlich die Schleife mit FOR..TO..NEXT zu programmieren.

Wir wollen in **Version 11** des immer noch gleichen Programms dazu die Zeilen 30, 50 und 60 ändern:

30 FOR Z=1 TO 374
50 NEXT Z
60 entfällt

Lassen Sie's laufen. Huiiih! Das pfeift runter! Der VC 20 meldet 0,91 Sekunden, der C 64 1,08 Sekunden. Das bringt gegenüber der IF..THEN-Schleife der Version 10 eine ganze Menge, nämlich ungefähr 2 Sekunden, oder, auf die Zeit der Version 1 bezogen, 89 Prozent.

Das einzige, was wir in der FOR..NEXT-Schleife noch verbessern können, ist die vielgeübte Praxis des Weglassens der Schleifenvariable Z nach dem NEXT-Befehl.

50 NEXT

Der Geschwindigkeitsgewinn dieser **Version 12** ist nicht sehr groß, nämlich nur etwa 0,1 Sekunden, aber wir wollen die Methode doch in die nächste Regel mit aufnehmen.

Regel 5
Schleifen sollen nicht mit IF..THEN, sondern mit FOR..TO..NEXT gebildet werden. Die Schleifenvarible nach NEXT soll dann weggelassen werden, wenn es nicht zu Verwechslungen mit anderen Schleifen führen kann.

Ich bin fast am Ende meines Beschleunigungslateins. Nur eines bleibt noch, nämlich die bisher hochgehaltene Lesbarkeit des Programms zu opfern. Ich hoffe nämlich, Sie haben bisher meiner Eingangsforderung Folge geleistet und alles schön mit Leerzeichen geschrieben. Das behalten wir zunächst noch bei, im Gegenteil, wir wollen zunächst die Lesbarkeit noch erhöhen und REM-Erläuterungen einfügen. Ich schlage vor, die **Version 13** so auszuschmücken:

```
  10 TI$="000000":REM UHR AUF NULL
  12 REM ******************************
  13 REM*                             *
  14 REM*TEST-PROGRAMM*               *
  15 REM*                             *
  16 REM ******************************
  20 PRINT CHR$(147);:REM ALLES LOE-
     SCHEN
  30 FOR Z=1 TO 374:REM 374 ZEICHEN
  40 PRINT"A";
  50 NEXT
 999 REM ZEIT AUSDRUCKEN
1000 POKE 214,18:PRINT TI/60 "SEKUN-
     DEN":END
```

Sieht gut aus, nicht wahr?

Aber leider, REM-Erläuterungen kosten Zeit. Wir sind um 0,15 (0,2) Sekunden langsamer geworden.

Wir schmeißen deshalb alle REMs wieder raus und haben damit wieder Version 12. Jetzt aber gehen wir einen Schritt in der anderen Richtung weiter und entfernen alle Leerstellen und Abstände. Mit dieser **Version 14** will ich Ihnen zeigen, daß das auch einen Einfluß auf die Laufgeschwindigkeit hat.

```
  10 TI$="000000"
  20 PRINTCHR$(147);
  30 FORZ=1TO374
  40 PRINT"A";
  50 NEXT
1000 POKE214,18:PRINT:PRINT-
     TI/60"SEKUNDEN":END
```

Das Ergebnis ist für den VC 20 0,81 Sekunden, für den C 64 0,98 Sekunden.

In **Version 15** treiben wir die Schrumpfung ins Extrem, indem wir das Programm im Prinzip unverändert aber mit einem Minimum an Zeilen schreiben, also möglichst viele Befehle in eine Zeile packen.

Sie wissen, die maximale Zeilenlänge beträgt 88 Zeichen beim VC 20, 80 Zeichen beim C 64. Unser Programm können wir sogar in einer einzigen Zeile unterbringen — fast unglaublich, aber es geht. Sie müssen allerdings alle Abkürzungsmöglichkeiten ausschöpfen, die das Commodore-System bietet. Im Anhang der Commodore-Handbücher finden Sie die Liste aller Abkürzungen beim Eintippen: C und geSHIFTEtes H für CHR$, ? für PRINT und so weiter. Im nachfolgenden Ausdruck ist das natürlich nicht zu sehen, weil der Befehl LIST die Abkürzungen nicht berücksichtigt. So kommen auch mehr als 88 (80) Zeichen in eine Zeile des Listings, woran Sie sich nicht stören dürfen.

```
10 TI$="000000":PRINTCHR$(147);
:FORZ=1TO374:PRINT"A";:NEXT
:POKE214,18:PRINT:PRINT TI/60
"SEKUNDEN":END
```

Und siehe da, diese »Kurzform« des Programms ist auch die allerschnellste Version. Der VC 20 braucht 0,78 Sekunden, der C 64 0,93 Sekunden. Diese letzte Beschleunigung wird dadurch erreicht, daß das Betriebssystem des Computers nur einmal einen Zeilenanfang und Zeilenende suchen und erkennen muß, statt sechsmal in der Version 14.

Das Ausnützen der vollen Kapazität einer Zeile bringt also nicht nur den Vorteil eines kleineren Speicherbedarfs, sondern auch Zeitgewinn.

Regel 6
* Programme ohne REM-Erläuterungen und ohne Leerstellen zwischen den Zeichen laufen schneller.
* Zur Reduzierung der Zeilenzahl sollen möglichst viele Befehle in eine Zeile geschrieben werden.

»Einzeiler« können auch Spaß und Herausforderung zugleich sein. Man sollte eigentlich annehmen, daß mit einer Zeile nicht viel anzufangen sei.

Weit gefehlt!

Einzeilige Programme

Eine englische Zeitschrift hat darüber sogar einige Male einen Wettbewerb ausgeschrieben (die 64'er ist unabhängig davon auch auf diese Idee gekommen, siehe Ausgabe 8; d. Red.). Ich schreibe die Einzeiler unten lesbar, das heißt mit Leerstellen. Sie müssen die Programme aber wieder mit allen Abkürzungstricks schreiben, sonst geht's schief.

□Von A. Boyd (Manchester) stammt ein Primzahlen-Erzeuger, der für die obere Grenze von 65000 viele Stunden braucht.

```
1 FOR N=1 TO 65000:F=0FOR J=2
TO N—1:F=F+((N—J*INT(N/J))=0)
:NEXT:X=—(F=0):PRINT
RIGHT$(STR$(X*N),6*X):NEXT
```

□Ein anderer Einzeiler wurde von A.M. Simmelt (Sheffield) geschrieben zur Konvertierung von Dezimalzahlen in Dualzahlen.

```
1 INPUT A:FOR I=14 TO 0 STEP—1
:Z=A AND 2↑I:A=A—Z:Z=Z/2↑
I:Z$=RIGHT$(STR$(Z),1):PRINT Z$;:
NEXT:GOTO1
```

□Zuletzt noch ein Juwel, nur für den VC 20 geeignet, von M. Dooling (Dewsbury). Hochauflösende Grafik und Scrolling in einer Zeile — wer hält's für möglich?

```
1 POKE 36869,255: PRINT"[SHIFT
CLR/HOME]@@@@@@@@@@
@@":FOR X=7168 TO 7175:POKE
X, PEEK(X+1):NEXT:POKE7176,
PEEK(A+8↑5):A=A+1:GOTO 1
```

Lassen wir's gut sein mit diesem Programmsport und kehren wir zurück zu einer abschließenden Betrachtung der Zeitgewinne.

Wir haben in Version 1 mit 8,25 (10,48) Sekunden begonnen. Diese Laufzeit wurde ohne Änderung des Programmresultats stetig verkürzt, bis wir schließlich in Version 15 bei 0,78 (0,93) Sekunden gelandet sind.

Ich nenne diese Beschleunigung um 90 Prozent schlicht und einfach spektakulär.

Mehr allerdings kann ich nicht herausholen, es sei denn — na ja, eigentlich habe ich am Anfang ganz laut »Basic« gesagt.

Aber ich kann doch nicht widerstehen und Sie scharf machen auf ultima velocitas — zu deutsch Maschinensprache. Dazu aber in der nächsten Ausgabe.

(Dr. Helmuth Hauck/aa)

Datex-P und ausländische Netzwerke

Datex-P ist ein Daten-Vermittlungssystem der Deutschen Bundespost. In diesem Netz wird die Vermittlung der Daten in Form von sogenannten Paketen (daher auch das 'P' in Datex-P) vorgenommen.

Das Datex-P-Netz ist ein flächendeckendes, öffentliches Datenübertragungsnetz, welches auf der Technologie der Paketvermittlung basiert. Bei dieser Technik werden die zu übertragenden Daten an einen Netzknoten übermittelt, dort kurzzeitig zwischengespeichert und dann — eventuell über andere Knoten — an die Zieladresse weitergeleitet. Paketvermittlung ist dadurch charakterisiert, daß die zu sendenden Daten in Pakete maximaler Länge zerlegt werden, die mit zusätzlichen Verwaltungsinformationen versehen dem Netz zur Weitervermittlung zugeführt und schließlich vom empfangenden Teilnehmer wieder zusammengesetzt werden. Die Vermittlungsknoten sind durch digitale Leitungen hoher Geschwindigkeit miteinander verbunden. Die Beibehaltung der Reihenfolge gesendeter Pakete wird vom Paketvermittlungsnetz garantiert. Der Datenaustausch zwischen der sendenden und der empfangenden Endeinrichtung wird über eine logische Verbindung abgewickelt, bei der aber (im Gegensatz zur Leitungsvermittlung) die Übertragungsleitungen nicht exklusiv reserviert, sondern mehrfach genutzt werden, da sie nur zur tatsächlichen Transportzeit in Anspruch genommen werden. Man spricht in diesem Zusammenhang von einer virtuellen Verbindung.«

So lautet die offizielle Definition im Datex-P-Handbuch der Post.

Bild 1. Das Datex-P-Netz schematisch

»Leitungsvermittlung« verwendet zum Beispiel der Datel-Dienst Datex-L der Post, aber auch das normale Telefon. Man kann sich also das Ganze als ein Netz von einzelnen Computern vorstellen, die nichts anderes zu tun haben, als ankommende Daten (Pakete) an den richtigen Computer weiterzuleiten (Bild 1). Die »Adresse« des Zielcomputers entnehmen sie dabei der Kopfinformation des jeweiligen Pakets, das heißt am Anfang eines Datenpakets steht der »Empfänger« des Datenpakets. Beim Aufbau des ganzen Netzes liegt der Vergleich zum Telefonnetz nahe, auch hier besteht das Netz aus Vermittlungsstellen, deren Funktion mit der der »Vermittlungscomputer« des Datex-P-Netzes äquivalent ist.

Der Vorteil eines Netzwerkes liegt auf der Hand: Die Anzahl der Leitungen, die nötig sind, um eine Verbindung zwischen zwei beliebigen Datenendeinrichtungen möglich zu machen, ist erheblich geringer, als wenn jede Datenendeinrichtung mit jeder verbunden wäre.

Zugang zum Datex-P-Netz mit Heimcomputer oder Terminal

Man braucht für den Datex-P-Zugang generell drei verschiedene Dinge: Eine Tastatur, einen Bildschirm oder eine andere Anzeige, um die empfangenen Daten anzuzeigen und ein Modem (MODulator-DEModulator), der die Daten entsprechend umwandelt, damit sie über das Telefon übertragen werden können (Bild 2). Zum Begriff MODEM: Alle Vorrichtungen, die »Daten modulieren« sind der Definition nach Modems. Akustikkoppler sind dabei eine Untergruppe, welche die Information nicht durch direkte Verbindung mit dem Telefonnetz übertragen, sondern mit Lautsprecher und Mikrofon eine »akustische Koppelung« erreichen. Als normaler Benutzer des Datex-P-Netzes braucht man sich um die Paketübertragung nicht zu kümmern, da die Umwandlung der Daten der angewählte PAD (Packet Assembly Disassembly Facility), auch wieder ein Computer, übernimmt. Dabei kommen die Daten in Form von einzelnen Zeichen an, werden vom PAD in Datenpakete umgewandelt und an die richtige Adresse weitergeleitet. Genauso werden empfangene Datenpakete vom PAD in einzelne Zeichen umgewandelt. Bei Datex-P-Betrieb mit Terminal ist in der Regel außer einem Modem und einem Telefon nichts weiter erforderlich. Bei einem Heimcomputer braucht man, damit er eine Terminalfunktion übernimmt, ein Terminalprogramm (wie zum Beispiel das Teleterm für den C 64) und, wenn er die erforderliche Schnittstelle zum Modem nicht bietet, noch ein Interface, das die nötige Umwandlung der Daten übernimmt. Dann kann es losgehen, man wählt, je nach Übertragungsrate des Modems (in der Regel hat man ein 300-Baud-Modem) die entsprechende Nummer des nächsten PADs an.

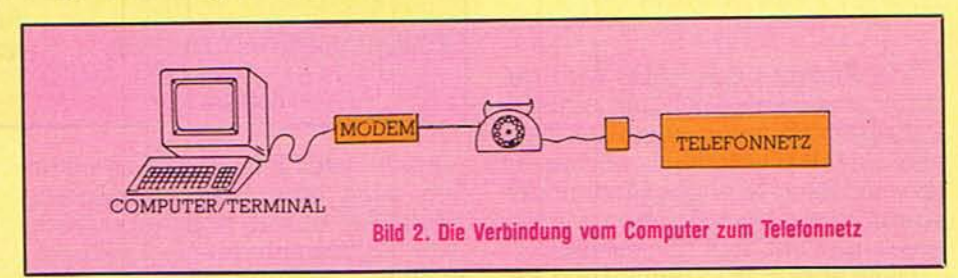


Bild 2. Die Verbindung vom Computer zum Telefonnetz

Tabelle 1 gibt eine Übersicht der in Deutschland anwählbaren PADs.

Jeder Commodore 64-Benutzer, der in der Lage ist, Mailboxen wie Decates oder Tedas anzuwählen, ist mit der selben technischen Einrichtung auch in der Lage, die PADs anzuwählen, um am Datex-P-Netz teilzunehmen (300-Baud-Nummern!).

Es besteht auch die Möglichkeit, einen Datex-P-Hauptanschluß zu beantragen. Dann erhält man eine Extra-Leitung, über die die Daten im Datex-P-Paketformat ankommen. Die billigste Version kostet 300

PAD	Vorwahl	300 Baud	1200 Baud	1200/75 Baud
Augsburg	0821	464011	464031	464051
Berlin	030	240001	240081	240061
Bielefeld	0521	59011	59021	59041
Bremen	0421	310131	314291	315077
Dortmund	0231	57011	52011	52081
Düsseldorf	0211	329318	329249	320748
Essen	0201	787051	791021	793003
Frankfurt	069	20281	20291	20201
Hamburg	040	441231	44261	441281
Hannover	0511	326651	327481	327591
Karlsruhe	0721	60241	60381	60581
Köln	0221	2911	2831	2951
Mannheim	0621	39931	39941	39951
München	089	228730	228630	228758
Nürnberg	0911	20571	20541	20501
Saarbrücken	0681	810011	810031	810061
Stuttgart	0711	299171	299061	299291

Achtung: Die Vorwahl für Frankfurt lautet ab 5.8.84 »069«

Tabelle 1. Eine Liste der in Deutschland anwählbaren PADs (Packet Assembly Disassembly Facility)

Mark im Monat und dürfte für keinen, der einen Homecomputer besitzt, in Frage kommen. Die Computer, die an einem solchen Hauptanschluß hängen, sind dann die, die »angerufen« werden können.

So wird eine Verbindung hergestellt

Die Nummer des PADs wird gewählt, das Telefon mit dem Akustikkoppler verbunden, wenn das Piepsen des PAD-Modems hörbar ist (bei 300 Baud wird das Modem des Anrufers wie beim Anrufen von Decates auf ORIGINATE gestellt).

Dann tippt man ».« (Punkt) und RETURN gegebenenfalls so lange, bis sich der PAD mit »Datex-P« und der Nummer seines Anschlusses meldet. Jetzt kann sich der Benutzer durch Eingabe seiner Benutzerkennung (NUI) identifizieren und/oder eine sogenannte NUA (Network User Adress) eingeben. Das ist nichts anderes als eine Art »Telefonnummer« innerhalb des Datex-P-Netzes). Nach Eingabe der NUA meldet Datex-P, ob der angerufene Anschluß frei ist (»Verbindung hergestellt«), besetzt ist (»Gegenstelle belegt«) oder nicht antwortet (»Gegenstelle antwortet nicht«). Wie beim Telefon gibt es eine Meldung »kein Anschluß unter dieser Nummer«.

Beim Datex-P-Netz gibt es natürlich auch eine Gebührenabrechnung. Sie richtet sich dabei nicht nach Entfernung der Verbindungspunkte, sondern hauptsächlich nach der Menge der übertragenen Daten.

Was ist eine NUI?

Das Abrechnungssystem des Datex-P-Netzes basiert auf sogenannten NUIs (Network-User-Identifications), die den Teilnehmer identifizieren, der für die Verbindung zu zahlen hat. Dieses Abrechnungssystem hat den Vorteil, daß der Datex-P-Benutzer von jedem Telefon aus in das Netz gelangen kann. Die NUI besteht insgesamt aus zwei Teilen, Teil A und B. Bei der Beantragung der NUI sucht sich der Teilnehmer den Teil A der NUI selbst aus, den Teil B der NUI, das Passwort, bekommt er dann von der Post schriftlich mitgeteilt. Dieser Teil B wird mittels eines bestimmten Algorithmus' aus dem Teil A der Teilnehmererkennung errechnet.

Ohne eine NUI erreicht man nur Teilnehmer, deren Anschluß die Gebühren übernimmt, und das auch nur innerhalb Deutschlands. Bei der Anfrage auf Gebührenübernahme muß vor der NUA die Zeichenfolge »R« und » « (Leerzeichen) stehen. Übernimmt der angewählte Anschluß die Gebühren nicht, so meldet der Datex-P-Computer (PAD) »Gebührenübernahme verweigert«. Nun eine Liste von Computern, die ein Benutzer ohne Eingabe einer Benutzerkennung über Datex-P erreichen kann (jede einzelne Nummer ist getestet):

Eingabe nach Meldung des PADs	Verbindung mit
R 45890040207 <RETURN>	IABG-Cyber (Großrechner) Techn. Dienstleist. und Beratungen für öff. Ber.
R 45211040026 <RETURN>	PRIMENET Stadt Düsseldorf
R 45611040037 <RETURN>	Control Data Corporation Hersteller von Großrechner.
R 45221040002 <RETURN>	DIMDI — Informationsdienst für Medizinische Bereiche
R 45611040250 <RETURN>	Gateway zum Tymnet-Amerikanischen Netzwerk
R 45231040017 <RETURN>	noch einmal Primenet

Tabelle 2. Liste von Computern ohne Eingabe einer Benutzerkennung

Kurz zusammengefaßt: Im normalen Zustand des PADs, wenn er also eine Eingabe verlangt, kann man immer eine Nummer »wählen«, also eingeben. Hat man sich durch Eingabe seiner NUI identifiziert, ist die Eingabe jeder Nummer möglich, hat man sich nicht identifiziert, kann man nur die »reverse Charging«-Nummern, also die, die die anfallenden Gebühren übernehmen, anwählen. Die Anfrage auf Gebührenübernahme durch die Gegenstelle geschieht dadurch, daß die Zeichen »R« und » « (Leerzeichen) vor der Nummer zu stehen haben. Sonst sind folgende Befehle wichtig:

Hat der PAD eine Verbindung zu einem anderen Computer hergestellt, und möchte dem PAD einen Befehl übermitteln, drückt man generell CTRL-P. Dann wird der Befehl eingegeben, und dieser mit RETURN abgeschlossen. Hat der PAD

keine Verbindung aufgebaut, ist die Eingabe von CTRL-P nicht erforderlich.

CLEAR

Dieser Befehl bricht die bestehende Verbindung ab. In der Praxis wird er dann angewandt, wenn man nicht weiß, wie man den Computer normal verlassen kann. Dieser Befehl hat also nur dann einen Sinn, wenn Datex-P eine Verbindung aufgebaut hat.

NUI Dxxxxxxxxx

Der Benutzer muß sich identifizieren, das heißt er gibt seine Kennung ein (Teil A). Der PAD fragt danach den Teil B der NUI (das Passwort) ab. (die »x« stehen hier für bei der Beantragung der NUI frei wählbare Zeichen).

SET 3:126, 5:1, 12:1

Diese Befehlsfolge setzt im PAD einige Parameter auf bestimmte Werte. Damit kann das Terminalprogramm mit dem sogenannten X-ON/X-OFF-Protokoll fahren, das heißt der PAD stoppt die Ausgabe der Daten, wenn er ein CTRL-S empfängt und fährt wieder damit fort, wenn er ein CTRL-Q erhält. Teleterm arbeitet zum Beispiel mit dieser Sequenz, um Daten auf Diskette abzuspeichern.

NUI OFF

Hiermit wird die korrekt eingegebene NUI gelöscht, das heißt jetzt kann eine andere eingegeben werden. Die Post empfiehlt diesen Befehl vor Auflegen des Telefons. Damit ist sichergestellt, daß die NUI nicht im PAD gültig bleibt, und von anderen mißbraucht werden kann.

Wie beantragt man eine NUI, was kostet sie?

Der beste Weg zu einer eigenen NUI: Man ruft das örtliche Fernmeldeamt der Post an und läßt sich die Nummer der für Datex-P zuständigen Stelle geben. Dort sagt man einfach, man möchte eine NUI beantragen, und sie sollen ein Antragsformular zuschicken (auch nach einem Freiumschlag fragen, dann braucht man kein Porto zu zahlen). Auf dem Antragsformular steht dann alles weitere. Dieses Antragsformular steckt man ausgefüllt in den Umschlag, und nach ungefähr einer Woche bekommt man von der Post Nachricht, daß man den Teil B der NUI, das Passwort, beim nächsten Postamt mit seinem Ausweis abholen kann. Bis hierher entstehen einem keinerlei Kosten. Ab da kostet die NUI monatlich 15 Mark, ohne daß sie benutzt werden muß (ähnlich der Grundgebühr beim Telefonanschluß). Die Verbindungskosten berechnen sich hauptsächlich nach der Menge der übertragenen Daten. Eine angewählte Verbindung zum PAD (300 Baud) kostet:

Der PAD-Zugang 4 Pfennig pro Minute;

jede bereitgestellte Verbindung 5 Pfennig;

Der Datex-P Gebrauch 1 Pfennig pro Minute.

Die Übertragung von 64 Bytes kostet zusätzlich:

von 8 bis 18 Uhr 0,33 Pfennig
von 6-8 und 18-22 Uhr 0,22 Pfennig
von 22-6 Uhr 0,11 Pfennig

Diese Preise senken sich mit der übertragenen Anzahl von 64-Byte-Gruppen; es gibt also eine Art Mengenrabatt. Bei Verbindungen mit dem Ausland wird noch eine zusätzliche Zeitgebühr berechnet, die zum Beispiel für die USA 0,20 Pfennig pro Minute beträgt.

Lohnt sich eine NUI?

Diese Frage muß jeder für sich selbst beantworten, vor allem dann, wenn er sie nicht geschäftlich, sondern privat nutzen will. Der wesentliche Vorteil von Datex-P ist: Eine Verbindung über den Bereich des örtlichen Telefonnetzes hinaus kommt über Datex-P immer billiger, als wenn man direkt dort anrufen würde. Natürlich hängt es von den finanziellen Verhältnissen ab. Viele jugendliche C64-Besitzer sind kaum in der Lage, eine NUI von ihrem Taschengeld zu finanzieren. Auch wird die Unterschrift des Anschlußinhabers verlangt, das heißt die Unterschrift der Eltern, die sie aus Angst vor den Kosten verweigern werden. Ich persönlich rate jedoch jedem, der es sich leisten kann, eine NUI zu beantragen. Mit einer NUI kann man zum Beispiel ein CBBS, ein Computer Bulletin Board Service in Aachen erreichen (NUA 44241040341), das sogenannte RMI-Net. Für die Benutzung dieses Services werden keinerlei weitere Gebühren verlangt, auch nicht (im Gegensatz zu Decates) wenn man eingetragener Benutzer werden will. Ich habe den Betreibern des RMI-Net einige Fragen gestellt, die der SYSOP (Systemoperator) des RMI-Net, Rupert Mohr, wieder mit einer Message beantwortete. Mich interessierten die Beziehungen zwischen dem RMI-Net und dem Chaos-Computer-Club, ob sich seiner Meinung nach eine NUI für einen Privatmann lohnt, warum sie nicht mehr ohne NUI erreichbar sind, und ob es etwas ähnliches wie RMI noch einmal gibt. So sehen die Nachrichten im Originaltext am Bildschirm aus:

Msg # ★840 Dated 07-10-84 11:13:58
From: SYSOP
To: THOMAS OBERMAIR
Re: (R)Fragen-2
1. Zwischen mir und dem CCC bestehen keine Beziehungen.
2. Ich kann nicht beurteilen, ob sich für einen Privatmann eine NUI lohnt. Im Raum Aachen lohnt sich aber unter Umständen schon ein Hauptanschluß für einen Privatmann.
3. Reverse Charge kostet Geld, unser Geld.
4. Das kommt auf die Definition von ähnlich an. Wenn Du gebührenfrei meinst, dann: NEIN.
Weiter (J), N, NS, RE, K? k
Msg # 840 Killed.
Msg # (577 to 842, ★, <H>elp)?

RMI ist ein Unternehmen, das Software für den IBM-PC herstellt. Das CBBS läuft ebenfalls auf einem IBM-PC mit Harddisk. RMI hat Zugang zu verschiedenen Datenbanken (wie Compuserve, The Source) und sammelt dort Informationen, die in ihrem CBBS öffentlich zugänglich gemacht werden. Sie bietet RMI eine umfangreiche Programmbibliothek an. Wenn genügend C 64-Anwender RMI benutzen würden, könnte man eine spezielle Rubrik für C 64-Programme einrichten lassen.

Der Zugang zu ausländischen Netzwerken

Da verschiedene Länder Datenvermittlungsnetze auf Paketbasis anbieten, liegt der Zugang von einem Netz zum anderen nahe. Dies geschieht durch sogenannte Auslandsvermittlungsstellen, die die eventuell verschiedenen Protokollformate untereinander anpassen (siehe Bild 3). Ferner übernehmen sie dieselben Aufgaben, wie die Vermittlungscomputer des Datex-P-Netzes. In der Praxis sieht das so aus: Jedes Netz hat eine eigene »Vorwahl«, die, wie beim Telefon, vor der eigentlichen Teilnehmer-Nummer eingegeben wird. Datex-P verlangt vor jeder

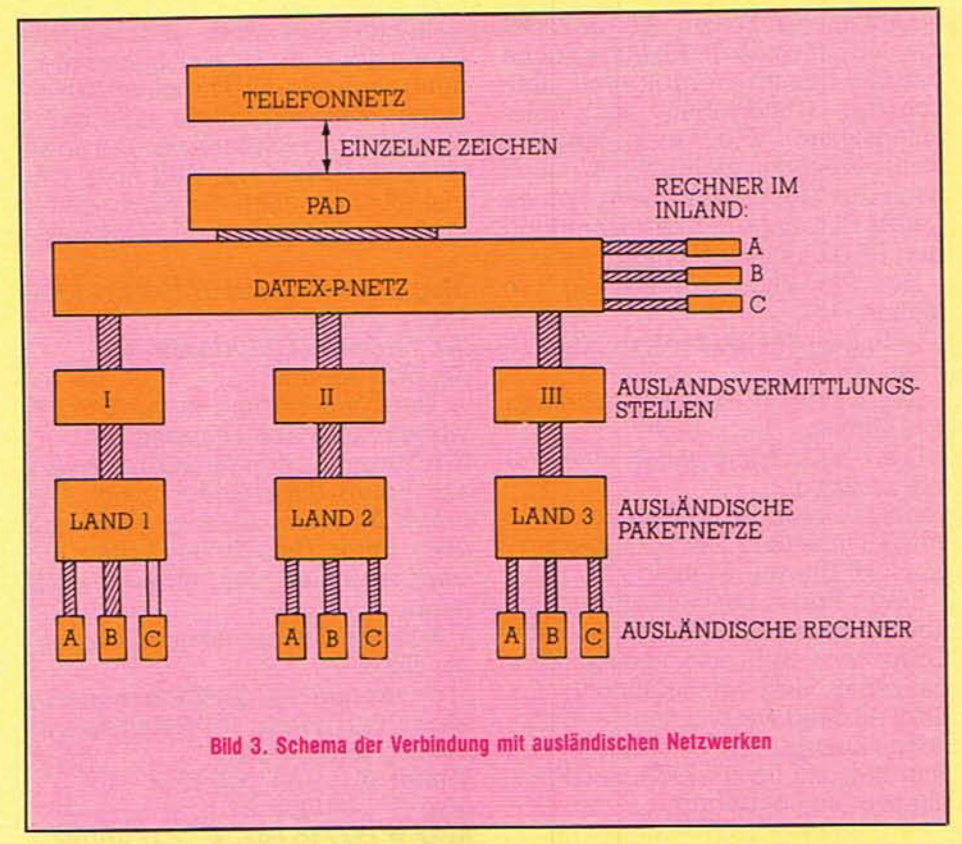


Bild 3. Schema der Verbindung mit ausländischen Netzwerken

Auslandsnummer die Ziffer 0 (Null) als Kennzeichen für einen Verbindungswunsch mit dem Ausland. In Tabelle 3 sehen Sie eine nicht vollständige Aufstellung erreichbarer Netze mit ihren »Vorwahlen« (sogenannte Netzkennzahlen). Das »in ausländische Netze eindringen« ist also meistens nichts anderes, als das Wählen einer vorher bekannten NUA im Ausland. Mit folgender Ausnahme: Es gibt Computer, die einen Anrufer weitervermitteln und einen recht komfortablen Dialog bieten. Generell gilt: Für Verbindungen mit dem Ausland braucht man eine NUI. Auslandsnummern findet man in größeren Mailboxen, wie in Decates oder im RMI-Net (über Datex-P). Soweit man die Datenbanken nicht bezahlen will oder kann, muß man Hacken, das heißt die verlangten Passwörter herausfinden. Wertvolle Kontaktadresse hierzu: Der Chaos Computer Club, ein Club von Computerfreaks, die es sich zur Philosophie gemacht haben, möglichst auf alles kostenlosen Zugriff zu haben. Erreichbar sind sie unter der RMI-Net-Mailbox mit dem Empfängernamen CHAOS TEAM, oder über normalen Postweg an folgende Adresse: Chaos Computer Club c/o Schwarzmarkt Bundesstr. 9 D 2000 Hamburg 13. Es lohnt sich, das Informationsblatt »Die Datenschleuder«, die unregelmäßig erscheint, im Abonnement kommen zu lassen. Das kostet etwa 30 Mark für ein Jahr.

Land	Netz	Zahl
Belgien	DCS	2062
	DCS	2068
	DCS	2069
	EURONET	2063
Dänemark	DATAPAK	2382
	EURONET	2383
England	IPSS	2341
	PSS	2342
	EURONET	2343
Kanada	DATAPAK	3020
	GLOBEDAT	3025
USA	TYMNET	3106
	TELENET	3110
	DATAPAK	3119
	AUTONET	3126
Japan	DDX-P	4401
	VENUS-P	4408
Italien	ITAPAC	2222

Tabelle 3. Liste der über Datex-P erreichbaren ausländischen Netze (nicht vollständig) mit der »Vorwahlnummer«

Mittlerweile sind bereits drei Ausgaben erschienen.

Zum EURONET: Vor einigen Jahren haben die Mitgliedstaaten der EG beschlossen, ein gemeinsames Netz von Paketvermittlungsdiensten aufzubauen. Ein einziger Großrechner übernimmt für alle Länder der EG die Vermittlungsaufgaben. Diesen Computer soll es in einem Jahr nicht mehr geben; das heißt die Vermittlung der Datenpakete wird, wie jetzt schon teilweise realisiert, über »normale« Auslandsvermittlungsstellen durchgeführt.

Was bringt die Zukunft?

Die Post plant für die nähere Zukunft ein »einheitliches digitales Übertragungsnetz« (ISDN). Dieses Netzwerk vereinheitlicht alle Dienste der Post in einem Übertragungsnetz, das heißt sowohl Sprache als auch Datenübertragung jeder Art (Daten, Texte, digitalisierte Bilder etc.) werden über ein Netz vermittelt. Dieses Netz soll bereits 1987 verfügbar sein. Dann soll vom ISDN zum IBFN (integriertes breitbandiges Fernmelde-Netz) allmählich übergegangen werden. Das IBFN unterscheidet sich vom ISDN im Prinzip nur durch die Übertragungsmedien. Daten werden dann über Glasfaser und Satelliten übertragen. Ein Anschluß zu einem Netzwerk wird so alltäglich, wie das heutige Telefonnetz. Das 64'er-Magazin wird Sie mit der Serie »Aktuelles aus der Datenfernübertragung« stets auf dem Laufenden halten.

Es ist mir leider nicht gelungen, Adressen von C 64-User-Clubs zu bekommen, die sich hauptsächlich mit Datenfernübertragung beschäftigen. Es existieren nur vereinzelt ein paar »Hacker«, die den C 64 fleißig beanspruchen. Wer Lust hat, einen solchen zu gründen, kann mir schreiben, oder mir eine Nachricht im RMI-Net hinterlassen. Ich werde dann ein solches Vorhaben nach meinen Kräften unterstützen.

(Thomas Obermair/aa)

Literaturliste zum Thema

MC-Modem-Sonderheft Franzis Verlag, Preis 16 Mark.
Dieses Sonderheft enthält alles Wissenswerte über Modems, Schnittstellen etc. Etwas zu teuer, aber es gibt nichts Vergleichbares, was billiger ist.

Elektronik-Sonderheft »Datenkommunikation« Franzis Verlag, 18 Mark.
Ein vom Informationsgehalt sehr hochwertiges Heft mit Fachbeiträgen von kompetenten Autoren. Fast zu kompliziert für den Laien, viel Theorie.

HYPRA-LOAD: Schnelles Laden von Diskette

Spontane Begeisterung und Zustimmung erhielten die beiden Autoren des folgenden Programms. Ein bis zu sechsmal schnelleres Laden von Programmen von der VC 1541 hatte uns bisher noch niemand angeboten. Wir meinen, ein echter Knüller, der es ganz bestimmt verdient, Listing des Monats zu werden.



Bisherige Versuche, die VC 1541 schneller zu machen, scheiterten fast alle. C 64-Besitzer hatten lediglich die Möglichkeit, mittels einer zusätzlichen IEEE-Schnittstelle eine der größeren Floppys (4040, 8250, etc.) anzuschließen. Viele waren der Meinung, daß die serielle Datenübertragung vom C 64 an die VC 1541 schuld an deren Trägheit hatte. Was natürlich nicht ganz stimmen kann. Denn es gibt serielle Schnittstellen, die mehr als 12000 Baud (= Bit pro Sekunde) übertragen. Somit war klar, daß nicht die serielle Übertragung, sondern eher das DOS (Disk Operating System), die interne Software der Floppy, der große Hemmschuh sein mußte. Und in der Tat, wenn Sie in der Ausgabe 9/84 des 64'er-Magazins auf Seite 10 die Aktuell-Meldung über eine neue Alternativ-Floppy gelesen haben, konnten Sie erfahren, daß lediglich das kompatible DOS die mögliche Geschwindigkeit des TEAC-Laufwerkes stark einschränkte. Und hier setzten die beiden Autoren mit ihrer Arbeit an. (gk)

Listing und Programmbeschreibung finden Sie auf Seite 77

Zusammenarbeit und Ehrgeiz werden mit 2 000 Mark belohnt

Ich, Boris Schneider, wurde am 5.7.66 in Grevenbroich (zirka 20 km südwestlich von Düsseldorf) geboren.

Genau 10 Jahre später erhielt ich meinen ersten Taschenrechner. Um 1978/79 hatte ich meine ersten Kontakte mit einem »richtigen« Computer: einem PET 2001, den mein Vater manchmal mit nach Hause brachte.

1980 erfolgte dann der Umzug nach Bayern; hier besuche ich das Gymnasium Vaterstetten, wo ich unter anderem auch Karsten Schramm kennenlernte, der sich nun ebenfalls kurz vorstellt:

Meine Geschichte begann am 1.3.66 in München.

Ich besuche zur Zeit ebenfalls das Gymnasium Vaterstetten, wo Boris und ich nächstes Jahr das Abitur machen wollen.

Mit 14 Jahren kam ich das erste Mal mit Computern in Kontakt, wobei ich prompt vom »Fieber« angesteckt wurde, so daß meine Eltern ernstlich um meine Zurechnungsfähigkeit besorgt sein mußten.

Meine Krankheit klang zum Glück schon nach einem Jahr ab, doch das Hobby blieb, und so besitze ich seit einem halben Jahr einen C 64, nachdem ich zuvor in der Schule auf CBM-Computern gearbeitet hatte.

Wir (Boris und ich) teilen uns die Leidenschaft, in Maschinensprache zu programmieren und dem Computer die letzten Geheimnisse zu entlocken!

Boris Oliver Schneider, Sudetenstraße 46, 8018 Grafing, Tel. 08092/6872.
Karsten Oliver Schramm, Herzog-Stephan-Weg 33, 8011 Zorneding, Tel. 08106/20252

Menügesteuerte Finanzmathematik

Klaus Klöker der Autor des Programms »Finanzmathematik«

Zwei herausragende Eigenschaften machen dieses Programm für alle C 64-Besitzer interessant: Die sehr gute Menütechnik und der modulare Bildschirmmaskenaufbau.

Wer wünscht sich nicht, daß Programme von Diskette auf Tastendruck geladen, beziehungsweise nachgeladen werden? Wenn Sie das Menüprogramm dieses Listings übernehmen und für Ihren Bedarf abändern, ist das Laden von Programmen kein Problem mehr.

Das Programm wird geladen mit dem Befehl:
LOAD "Menü" ,8
Gestartet wird dann mit dem Befehl: RUN

Wählen Sie nun mit den Funktionstasten F3 und F5 die Nummer des Programmes, mit welchem Sie als nächstes arbeiten wollen. Sie erkennen Ihre augenblickliche Position an der blinkenden Zahl.

Nach beendeter Wahl drücken Sie »RETURN«. Dann wird das Programm geladen, dessen Zahl gerade blinkt und an der sich nach »RETURN« ein ← (Pfeil links) befindet.

Bedienungsanleitung für das Programm:
Erstellen eines Tilgungsplanes.

Nach erfolgter Auswahl im Hauptmenü wird das Programm automatisch gestartet.

Durch entsprechende Informationen in der Fußzeile erklärt sich das Programm im wesentlichen selbst. Allerdings mußte manchmal wegen Platzmangel darauf verzichtet werden.

Grundsätzlich gilt:
— Bis auf die Funktionstasten erfolgt bei jeder Eingabe und Auswahl ein Abschluß mit »RETURN«.
— Bei der Auswahl im Programm, wird immer mit den Tasten F3 und F5 gewählt, nach Drücken von »RETURN« wird immer das als nächstes ausgeführt, was zuletzt geblinkt hat.
— Zum Abschluß der Dateneingabe gilt für die Abfrage, ob die Eingabe korrekt ist, folgendes:
a) Zugelassen sind die Buchstaben j (ja) und n (nein).
b) Die Antwort kann bis zum Abschluß mit »RETURN« durch die Taste »DEL« korrigiert werden.
— Während der Dateneingabe, kann man durch Drücken der Taste F1 zum ersten Auswahlmenü zurückkehren.
— Wird die Tabelle auf dem Bildschirm ausgedruckt, kann man den Bildschirmausschnitt, innerhalb der festgesetzten Grenzen verändern.

Mit F1 kehrt man zur alten Bildschirmmaske zurück.

Das Ausgabefeld »Resultat« enthält folgende Informationen:
— Zinsaufwand (gesamt): Dieser Summe entsprechen sämtliche in den einzelnen Jahren zu zahlende Zinsen.
— Annuität (gesamt): Dies ist die Summe, die man insgesamt zurückzahlen muß (Zins- und Tilgungsanteil zusammen).

Im linken unteren Feld wird noch einmal angege-

	Programmdokumentation: Menü
200 - 260:	Aufbau der Bildschirmmaske
300 - 390:	Auswahlroutine durch Betätigen der Funktionstasten
400 - 410:	Sprung in die Zeile, in der das entsprechende Programm geladen, oder in der das Programm beendet wird
900 - 960:	Festlegung der Variablen
	Variablenliste: Menü
na$ (...) =	Nummern der Programme
nb$ (...) =	Namen der Programme
Z =	Zeile
S =	Spalte
Y =	Index der indizierten Variablen
g$ =	Eingabevariable der get-Anweisung
	Programmdokumentation: Zinsrechnung
10 - 30	Unterprogramm zum Löschen des rechten oberen und linken unteren Bildschirmfensters
70 - 75	Unterprogramm zur Rundung der Ergebnisse und zur Umwandlung in einen String
80	Unterprogramm zum Laden des Menüs
90	Unterprogramm zum Setzen des Cursors, an die Stelle, welche mit Z = Zeile und S = Spalte vor dem Sprungbefehl festgelegt wurde
95	Unterprogramm zum Zeichnen einer Linie
100 - 195	Auswahlroutine mit den Funktionstasten der entsprechenden reversen Felder oben links und unten links.
200 - 290	Auswahlroutine mit den Funktionstasten der Fußzeile.
300 - 480	Eingaberoutine
1000 - 1180	Beginn des Hauptprogramms: — Aufbau der Bildschirmmaske — Sprung in die Auswahlroutine, um festzulegen, was berechnet werden soll.
2000 - 2140	Aufbau der Eingabemaske und Eingabe der Daten mittels der Eingaberoutine. Anschließend erfolgt Abfrage, ob die Eingabe korrekt ist.
3000 - 3040	Berechnung des vorher ausgewählten Ergebnisses
3500 - 3610	Ausdruck des Ergebnisses. Es werden ausgedruckt: — die am Anfang festgelegten Kriterien — die eingegebenen Daten — das Endergebnis

ben, wie hoch der Zinsanteil von der Annuität in Prozent ist. Dieses Programm stellt Tilgungspläne bis zu einer Laufzeit von 50 Jahren auf. Falls dies nicht ausreicht, braucht in Zeile neun nur die Dimensionierung der Variablen e und e$ erhöht werden. (Klaus Klöker/rg)

Lebenslauf

Ich bin 20 Jahre alt, habe im Frühjahr dieses Jahres mein Abitur bestanden und im August eine Ausbildung als Bankkaufmann begonnen.

In meiner Freizeit beschäftige ich mich gerne mit Geschichte und der Computerprogrammierung.

Entstanden ist das Programm aufgrund eines Kurses über Finanzmathematik.

Es schien mir damals zweckmäßig, Tilgungspläne mit dem Computer aufzustellen. Dieser absolviert seine Arbeit sekundenschnell und fehlerfrei. Wer selbst einmal Tilgung »zu Fuß« geplant hat, weiß, daß gerade letzteres keine Selbstverständlichkeit

Entstehungsgeschichte und Anwendung

ist (bei 20 Jahren Laufzeit und 6 Beträgen pro Jahr, müssen immerhin 120 Beträge einzeln berechnet werden).

Einen großen Teil der Programmierzeit beanspruchte der Aufbau der Bildschirmmasken und die Menüsteuerung. Ich hielt diesen Aufwand für angebracht, weil die Anwendung bequemer ist und außerdem mehr Spaß macht. Wer sich das Programm »Tilgungspläne« genauer anschaut, wird feststellen, daß man durch einfaches Austauschen der Formeln verschiedene Tabellen aufstellen kann, obwohl nur eine begrenzte Zahl von Eingabekriterien notwendig ist.

(Klaus Klöker)

	Es folgt: Aufbau der Fußzeile mit der Möglichkeit durch Wahl mit den Funktionstasten entsprechend zu verzweigen.
	Variablenliste: Zinsrechnung
ma$:	Kopfzeile
mb$, mc$, md$:	Fußzeilen
a$ (...):	Eingabekriterien
ka$; p$; zi$	
ja$; mo$; ta$	Ein- und Ausgabekriterien
pm$; pu$:	Punkte in der Eingabemaske
Z:	Zeile der Cursorposition
S:	Spalte der Cursorposition
Z1:	niedrigste Zeile, die gelöscht werden soll
Z2:	höchste Zeile, die gelöscht werden soll
Y:	Variablenindex
Y1:	kleinster Index
Y2:	größter Index
SW:	Schrittweite
k; c:	Konstanten, die, je nach ihrer Wertigkeit, bestimmte Vorgaben ausführen.
f:	gibt an, welche Berechnung erfolgen soll
e:	gibt an, die Art der Laufzeit
h:	legt die Art der Laufzeit in der Berechnung fest
in$:	Variable in der Eingaberoutine
ab$:	Endergebnis
	Programmdokumentation: Tilgungsrechnung
10 - 20	Unterroutine zum Löschen der rechten Bildschirmhälfte
30 - 55	Unterroutine zum Runden des Ergebnisses und zum Umwandeln in einen String
70	Unterroutine zum Laden des Menüs
80	Unterroutine zum Zeichen einer Linie
90	Unterroutine zur Festlegung der Cursorposition
95	Unterroutine zum Löschen der Fußzeile
100 - 195	Unterroutine zur Auswahl der vorgegebenen Möglichkeiten mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.
200 - 290	Unterroutine zur Auswahl bei der Fußzeile mit den Tasten F3 und F5, der Abschluß erfolgt mit »RETURN«.
300 - 480	Eingaberoutine. Durch Drücken der Taste F1 kann man von vorne beginnen
900 - 960	Aufbau der Bildschirmmaske
1010 - 1080	Ausdruck der Kopfzeile und Bildschirmmaske, Sprung in die Auswahlroutine (100-195)
2000 - 2140	Aufbau der Eingabemaske und Sprung in die Eingaberoutine. Zum Schluß wird abgefragt, ob die Eingabe korrekt ist. Die Antwort wird mit »RETURN« eingegeben und kann vorher mit »DEL« korrigiert werden.
3000 - 3720	Ausführung der Rechnung mit Sprung in die Rundungs- und Stringumwandlungsroutine
3800 - 3900	Ausdruck des Resultates auf dem Bildschirm
3910 - 3950	Aufbau der Wahlfußzeile und Sprung in die Auswahlroutine (200-290)
4000 - 4590	Ausdruck der Tabelle und Verändern des Bildschirmausdrucks mit den Cursortasten: Mit F1 erfolgt ein Rücksprung
5000 - 5090	Änderung der eingegebenen Daten
5500 - 5660	Papierausdruck der eingegebenen Daten und der Tabelle
6000 - 6500	Festlegung der Variablen
	Variablenliste: Tilgungsrechnung
ma$	Kopfzeile
mb$, mc$, me$, mf$, mh$	Fußzeilen
md$	Löschzeile zum Löschen des Bildschirms
Z:	Zeile der Cursorposition
S:	Spalte der Cursorposition
Y:	Variablenindex, der einfach dimensionierten Variablen
Y1:	niedrigster Variablenindex
Y2:	höchster Variablenindex
Sw:	Schriftweite
g$:	Variable der GET-Anweisung
g:	ASC (g$)
i,j:	Variablenindizes der zweifach dimensionierten Variablen
c:	Konstante, von der bestimmte Bedingungen abhängen
f:	gibt das Ergebnis der Auswahlroutine an
h:	beschränkt die maximale Eingabelänge
in$:	Variablenindex
pns:	füllt den Ergebnisstring mit Punkten auf
is:	gibt das Jahr in der Tabelle an
tb$:	chr$(16): Tabulator
ta$:	chr$(10): Leerzeile
X1:	Variable die den kleinsten Index der Tabelle angibt
X2:	Variable die den größten Index der Tabelle angibt
	Variablen der Rechnung t1,t2,J2,r1,a1,Z1,Z2

```
70 REM BY KLAUS KLOEKER
80 GOTO100
90 POKE214,Z:POKE211,S:SYS58732:RETURN
100 REM * MENUE *
110 PRINT"": POKE53280,0:POKE53281,0:PO
KE53272,23:POKE788,52:GOSUB900
200 PRINT"      ─ ╮ ╱ ♠ ╱ ♦ ╲ ♠ │  │ ‾ ╲ ♠
 ╮ ╯     "
210 Z=4:S=6:GOSUB90:PRINT"╲ENUE :":Z=5:G
OSUB90:PRINT"-------"
220 Z=7:S=0:Y=1:FORI=1TO5:GOSUB90:PRINTN
A$(I);NB$(I)
230 Z=Z+2:Y=Y+1:NEXT
240 Z=18:GOSUB90:PRINT"            ◌AEHLEN
♦IE MIT  ─3  UND  ─5  !"
250 Z=20:S=7:GOSUB90:PRINT"─RUECKEN ♦IE
DANN ' ─┐ ╭─╱ '  !"
260 Z=23:S=0:GOSUB90:PRINT"━━━━━━━━━━━━━━━━
"
300 REM * AUSWAHLROUTINE *
310 Z=7:Y=1
320 GOSUB90:PRINT"          ":FORI=1TO75:NE
XT
330 PRINT"□";NA$(Y):FORI=1TO125:NEXT
340 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN320
350 IFG$=CHR$(13)THENGOSUB90:PRINT">";
:GOTO400
355 IFG$=CHR$(135)THEN380
360 Z=Z+2:Y=Y+1:IFZ>15THENZ=7:Y=1
370 GOTO320
380 Z=Z-2:Y=Y-1:IFZ<7THENZ=15:Y=5
390 GOTO320
400 ONYGOTO500,510,520,530,540
410 END
500 LOADNB$(1),8
510 LOADNB$(2),8
520 LOADNB$(3),8
530 LOADNB$(4),8
540 PRINT"":POKE53272,21:POKE788,49:END
900 REM * VARIABLEN *
910 NA$(1)="─ 1 ─ ":NA$(2)="─ 2 ─ ":
NA$(3)="─ 3 ─ "
920 NA$(4)="─ 4 ─ ":NA$(5)="─ 5 ─ "
930 NB$(1)="♦INSRECHNUNG":NB$(2)="♦INSES
ZINSRECHNUNG"
940 NB$(3)="─ISKONTIEREN EINES ╯APITALS"
950 NB$(4)="‾RSTELLEN EINES │ILGUNGSPLAE
NES":NB$(5)="┐ROGRAMMENDE"
960 RETURN
READY.
```

Listing »Menüprogramm«

```
7 POKE45,PEEK(174):POKE46,PEEK(175):CLR
8 REM BY KLAUS KLOEKER
9 CLR:DIMA$(15),A(15):GOTO1000
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
20 FORZ=Z1TOZ2:GOSUB90:PRINT"
                    ":NEXT:IFC=0THENRETURN
30 Z1=12:Z2=20:C=0:GOTO20
50 RETURN
70 A$(12)=STR$(INT(A(12)*100+.5)):REM *
RUNDEN
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+"."+R
IGHT$(A$(12),2):RETURN
80 LOAD"MENUE",8:REM *LOAD MENUE
90 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION
95 A=55869:FORI=1597TO1622:POKEI,100:POK
EA,1:A=A+1:NEXT:RETURN
100 REM * AUSWAHLROUTINE *
110 GOSUB90:PRINT"                ":FORI=1T
O75:NEXTI
120 GOSUB90:PRINT""A$(Y)"":FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDCTHENC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB90:PRINT">":
RETURN
160 IFG$=CHR$(135)THEN190
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE *:S=0
210 Z=23:GOSUB90:PRINT"                ";:FO
RI=1TO75:NEXT
220 GOSUB90:PRINT""BA$(Y)"";:FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB90:PRINT">";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+10:Y=Y+1:IFS>31THENS=0:Y=1
270 GOTO210
280 S=S-10:Y=Y-1:IFS<0THENS=30:Y=4
290 GOTO210
300 REM * EINGABEROUTINE *
310 GOSUB90
320 IN$=""
330 PRINT"";
340 GETG$:IFG$=""THEN340
350 G=ASC(G$):IFG=13THENGOTO460
370 IFG$<"0"ORG$>"9"THEN420
380 IN$=IN$+G$
390 PRINTG$;
400 IFLEN(IN$)>NTHENFORI=1TOLEN(IN$):PRI
NTCHR$(20);:NEXT:GOTO320
410 GOTO330
420 IFG=46THEN380
425 IFG$=CHR$(133)THEN9
430 IFG<>20THEN330
440 IFLEN(IN$)<1THEN330
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO390
460 PN$="":FORI=1TO(10-LEN(IN$)):PN$=PN$
+".":NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
999 :
1000 REM ** ZINSRECHNUNG **
1010 GOSUB5000:PRINTMA$
1020 A=55388:FORI=1076TO1906STEP40:POKEI
,101:POKEA,1:A=A+40:NEXT
1030 A=55736:FORI=1464TO1474:POKEI,100:P
OKEA,1:A=A+1:NEXT
1040 A=55749:FORI=1477TO1502:POKEI,100:P
OKEA,1:A=A+1:NEXT
1050 Z=23:S=0:GOSUB90:PRINTMB$
1060 Z=3:GOSUB90:PRINT"♠USWAHL :":PRINT"
──────────"
1070 Z=5:FORI=1TO5:GOSUB90:PRINT""A$(I)
"":Z=Z+1:NEXT
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=
1:C=0:GOSUB100:F=Y:IFF=5THEN80
1090 Z=13:S=0:GOSUB90:PRINT"└AUFZEIT IN"
:PRINT"‾‾‾‾‾‾‾‾‾‾‾‾"
1100 Z=15:FORI=6TO8:S=0:GOSUB90:PRINT""
```




```
A$(I)"▒":Z=Z+1:NEXT
1110 Z=15:Z1=Z:Z2=17:S=0:Y=6:Y1=Y:Y2=8:S
W=1:C=0:GOSUB100:E=Y
1120 IFE=6THENA$(11)=JA$:H=1
1130 IFE=7THENA$(11)=MO$:H=12
1140 IFE=8THENA$(11)=TA$:H=360
1150 IFF=1THENA$(9)=KA$:A$(10)=P$:K=1
1160 IFF=2THENA$(9)=ZI$:A$(10)=P$:K=1
1170 IFF=3THENA$(9)=KA$:A$(10)=ZI$:K=0
1180 IFF=4THENA$(9)=KA$:A$(10)=ZI$:A$(11
)=P$:K=2
2000 REM * EINGABEMASKE *
2010 Z=3:S=13:GOSUB90:PRINT"¯INGABE :":Z
=4:GOSUB90:PRINT"¯¯¯¯¯¯¯¯¯¯"
2020 Z=5:FORI=9TO11:S=13:GOSUB90:PRINT"▒
"A$(I)"▒ : ";PM$
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB90:PRINTMC
$
2040 Z=5:S=27:N=10:GOSUB300:A(9)=VAL(IN$
)
2050 Z=7:S=27:N=10:GOSUB300:A(10)=VAL(IN
$)
2060 Z=9:S=27:N=10:GOSUB300:A(11)=VAL(IN
$)
2070 Z=13:S=13:GOSUB90:PRINT"¯INGABE KOR
REKT ? (J/N):":GOSUB95
2080 Z=13:S=38:GOSUB90:PRINT"▒▒";
2090 GETG$:IFG$<>"J"ANDG$<>"N"THEN2090
2100 PRINTG$:IFG$="J"THENY=1
2110 IFG$="N"THENY=0
2120 GETI$:IFI$<>CHR$(13)ANDI$<>CHR$(20)
THEN2120
2130 IFI$=CHR$(20)THEN2080
2140 C=1:GOSUB10:IFY=0THEN2010
3000 REM * RECHNUNG *
3010 IFF=1THENA(12)=(A(9)*A(10)*A(11))/(
100*H)
3020 IFF=2THENA(12)=(A(9)*100*H)/(A(10)*
A(11))
3030 IFF=3THENA(12)=(A(10)*100*H)/(A(9)*
A(11))
3040 IFF=4THENA(12)=(A(10)*100*H)/(A(9)*
A(11))
3500 REM * BILDSCHIRM
3510 Z=3:S=13:GOSUB90:PRINT"¯INGABE :":Z
=4:GOSUB90:PRINT"¯¯¯¯¯¯¯¯¯¯"
3520 Z=5:FORI=9TO11:GOSUB90:PRINTA$(I)"
:"A(I):Z=Z+2:NEXT
3530 Z=13:GOSUB90:PRINT"_ESULTAT :":Z=14
:GOSUB90:PRINT"¯¯¯¯¯¯¯¯¯¯":GOSUB70
3540 Z=16:GOSUB90:PRINT"▒"A$(F)"▒ :";:PR
INTTAB(38-LEN(AB$));AB$
3550 PRINTTAB(39-LEN(AB$));:FORI=1TOLEN(
AB$)-1:PRINT"=";:NEXT
3560 Z=20:GOSUB90:IFK=1THENPRINT"♠USDRUC
K IN ─\ !":GOTO3580
3570 IFK=0THENPRINT"♠USDRUCK IN ¯ROZENT
!"
3580 Z=23:S=0:GOSUB90:FORI=1TO4:PRINT"▒"
BA$(I)"▒";:NEXT
3590 Y=1:GOSUB200:IFY=4THEN80
3600 IFY=3THEN1010
3610 IFY=2THEN4500
4000 REM * AENDERUNG *
4010 C=1:GOSUB10
4020 Z=3:S=13:GOSUB90:PRINT"♠ENDERUNG :"
:Z=4:GOSUB90:PRINT"¯¯¯¯¯¯¯¯¯¯¯"
4030 Z=5:S=13:FORI=9TO11:GOSUB90:PRINT"▒
"A$(I)"▒ :";A(I):Z=Z+2:NEXT
4040 Z=23:S=0:GOSUB90:PRINTMD$;
4050 Z=5:Z1=Z:Z2=9:S=13:Y=9:Y1=Y:Y2=11:S
W=2:C=1:GOSUB100
4060 IFC=0THENGOSUB10:GOTO3000
4070 Z=13:S=13:GOSUB90:PRINT"▒"A$(Y)"▒ :
";PM$:GOSUB95
4080 Z=23:S=0:GOSUB90:PRINTMC$;:Z=13:S=2
7:GOSUB300:A(Y)=VAL(IN$):GOTO4010
4500 REM * AUSDRUCK *
4510 OPEN4,4,7:PRINT#4,CHR$(14);CHR$(16)
;"28♠,/♠_¯-|/,/!"
4520 PRINT#4,CHR$(16);"28------------";C
HR$(10)
4530 FORI=9TO11:PRINT#4,CHR$(15);CHR$(16
);"10*** "A$(I)" : ";A(I):NEXT
4540 GOSUB70:PRINT#4,CHR$(10);CHR$(16);"
14"A$(F)" : "AB$
4550 PRINT#4,CHR$(16);"15";:FORI=1TO2+LE
N(A$(F)+AB$):PRINT#4,"=";:NEXT
4560 PRINT#4:PRINT#4:PRINT#4
4570 PRINT#4,CHR$(13);CHR$(16);"10***
/APITAL UND ♠INSSATZ WERDEN";
4580 PRINT#4," IN ' ─\ ' AUSGEDRUCKT,"
4590 PRINT#4,CHR$(16);"16DER ♠INSSATZ IN
 ' % ' !"
4600 PRINT#4:FORI=1TO40:PRINT#4,"==";:NE
XT:PRINT#4:PRINT#4:PRINT#4:PRINT#4
4610 CLOSE4:FORI=56176TO56215:POKEI,0:NE
XT:GOTO3580
5000 REM * VARIABLENLISTE *
5010 MA$="▒▒        ♠ \ / ♠ _ ¯ - | / /
     ▒"
5020 MB$="▒ _3 U. _5 - ○AHL ! '_¬| _/' -
¯INGABE !▒"
5030 MC$="▒  _1 - /EU !     |EBEN ♠IE E
IN !     ▒"
5040 MD$="▒ _1 - _ESULTAT !      _3 U. _5
 - ○AHL ! ▒"
5050 KA$=" /APITAL  ":P$="  ♠INSSATZ ":
ZI$=" ♠INSERTRAG"
5060 JA$="  \AHRE   ":MO$="  \ONATE   "
:TA$=" |AGE     ":PM$=".........."
5110 BA$(1)=" ♠ENDERN  "
5120 BA$(2)=" /OPIE    "
5130 BA$(3)=" /.|ERECH."
5140 BA$(4)=" \ENUE    "
5510 A$(1)=" ♠INSERTRAG"
5520 A$(2)="  /APITAL  "
5530 A$(3)="  ♠INSSATZ "
5540 A$(4)="  LAUFZEIT "
5550 A$(5)="  \ENUE    "
5560 A$(6)="  \AHRE    "
5570 A$(7)="  \ONATE   "
5580 A$(8)="  |AGE     "
5590 A$(13)="  ♠ENDERN  "
5600 A$(14)="  _ESULTAT "
5610 A$(15)="  \ENUE    ":RETURN
READY.
```

Listing »Zinsrechung«

```
7 POKE45,PEEK(174):POKE46,PEEK(175):CLR
8 REM BY KLAUS KLOEKER
9 CLR:DIMA$(19),A(19),BA$(4):GOTO1000
10 Z1=2:Z2=10:S=13:REM * MASKELOESCHEN *
20 FORZ=Z1TOZ2:GOSUB80:PRINT"
         ":NEXT:IFC=0THENRETURN
30 Z1=12:Z2=20:C=0:GOTO20
40 S=0:FORZ=12TO20:GOSUB80:PRINT"
   ":NEXT:RETURN
50 A$(A)=STR$(INT(A(A)*100+.5)):REM * RU
NDEN
60 AB$=LEFT$(A$(A),LEN(A$(A))-2)+"."+RIG
```

```
HT$(A$(A),2):RETURN
70 LOAD"MENUE",8:REM *
80 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION
90 A=55869:FORI=1597TO1622:POKEI,100:POK
EA,1:A=A+1:NEXT:RETURN
100 REM * AUSWAHLROUTINE
110 GOSUB80:PRINT"▒                    ▓":FORI=1T
O75:NEXT
120 GOSUB80:PRINT"▒"A$(Y)"▓":FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDCTHENC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB80:PRINT"▒>▓";
:RETURN
160 IFG$=CHR$(135)THEN190
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE:S=0
210 Z=23:GOSUB80:PRINT"▒          ▓";:FO
RI=1TO75:NEXT
220 GOSUB80:PRINT"▒"BA$(Y)"▓";:FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB80:PRINT"▒>▓";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+10:Y=Y+1:IFS>30THENS=0:Y=1
270 GOTO210
280 S=S-10:Y=Y-1:IFS<0THENS=30:Y=4
290 GOTO210
300 REM * EINGABEROUTINE
310 GOSUB80
315 IN$=""
320 PRINT"▄▌";
330 GETG$:IFG$=""THEN330
340 G=ASC(G$):IFG=13THEN460
360 IFG<48ORG>57THEN410
370 IN$=IN$+G$:IN=LEN(IN$)
380 PRINTG$;
390 IFIN>NTHENFORI=1TOIN:PRINTCHR$(20);:
NEXT:GOTO310
400 GOTO320
410 IFG=46THEN370
420 IFG$=CHR$(133)THEN9
430 IFG<>20THEN320
440 IFIN<1THEN320
450 IN$=LEFT$(IN$,IN-1):GOTO380
460 PN$="":FORI=1TO(10-LEN(IN$)):PN$=PN$
+".":NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
999 :
1000 REM * ZINSESZINSRECHNUNG
1010 GOSUB5000:PRINTMA$
1020 A=55388:FORI=1076TO1906STEP40:POKEI
,101:POKEA,1:A=A+40:NEXT
1030 A=55736:FORI=1464TO1474:POKEI,100:P
OKEA,1:A=A+1:NEXT
1040 A=55749:FORI=1477TO1502:POKEI,100:P
OKEA,1:A=A+1:NEXT
1050 Z=23:S=0:GOSUB80:PRINTMB$
1060 Z=2:GOSUB80:PRINT"♠USWAHL :":PRINT"
──────────"
1070 Z=4:FORI=1TO5:GOSUB80:PRINT"▒"A$(I)
"▓":Z=Z+1:NEXT
```

Listing »Zinsrechnung« (Fortsetzung)

```
1080 Z=4:Z1=Z:Z2=8:S=0:Y=1:Y1=Y:Y2=5:SW=
1:C=0:GOSUB100:F=Y:IFF=5THEN70
1090 IFF=1THENA$(9)=KO$:A$(10)=P$:A$(11)
=N$:K=1
1100 IFF=2THENA$(9)=KN$:A$(10)=P$:A$(11)
=N$:K=1
1110 IFF=3THENA$(9)=KO$:A$(10)=KN$:A$(11
)=N$:K=0
1120 IFF=4THENA$(9)=KO$:A$(10)=KN$:A$(11
)=P$:K=2
2000 REM * EINGABEMASKE
2010 Z=2:S=13:GOSUB80:PRINT"¯INGABE :":Z
=3:GOSUB80:PRINT"──────────"
2020 Z=4:FORI=9TO12:S=13:GOSUB80:PRINT"▒
"A$(I)"▓ : ";PM$:Z=Z+2:NEXT
2030 Z=23:S=0:GOSUB80:PRINTMC$;
2040 Z=4:S=27:N=10:GOSUB300:A(9)=VAL(IN$
)
2050 Z=6:GOSUB300:A(10)=VAL(IN$)
2060 Z=8:GOSUB300:A(11)=VAL(IN$)
2065 Z=10:N=9:GOSUB300:A(12)=VAL(IN$)
2070 Z=13:S=13:GOSUB80:PRINT"¯INGABE KOR
REKT ? (J/N):":GOSUB90
2080 Z=13:S=38:GOSUB80:PRINT"▄▌";
2090 GETG$:IFG$<>"J"ANDG$<>"N"THEN2090
2100 PRINTG$:IFG$="J"THENY=1
2110 IFG$="N"THENY=0
2120 GETI$:IFI$<>CHR$(13)ANDI$<>CHR$(20)
THEN2120
2130 IFI$=CHR$(20)THEN2080
2140 C=1:GOSUB10:IFY=0THEN2010
3000 REM * RECHNUNG
3010 IFF=1THENA(16)=A(9)*(1+(A(10)/(A(12
)*100)))^(A(11)*A(12))
3020 IFF=1THENA(19)=100*((1+(A(10)/(A(12
)*100)))^A(12)-1):D=1
3030 IFF=2THENA(16)=A(9)/(1+(A(10)/(A(12
)*100)))^(A(11)*A(12)):D=0
3040 IFF=3THENA(16)=((A(10)/A(9))^(1/(A(
11)*A(12)))-1)*100:D=0
3050 IFF=4THENA(16)=(LOG(A(10)/A(9))/LOG
(1+(A(11)/(A(12)*100))))/A(12):D=0
3500 REM * SCREEN
3510 Z=2:S=13:GOSUB80:PRINT"¯INGABE :":Z
=3:GOSUB80:PRINT"──────────"
3520 Z=4:FORI=9TO12:GOSUB80:PRINTA$(I)"
:"A(I):Z=Z+2:NEXT
3530 Z=13:GOSUB80::PRINT"_ESULTAT :":Z=1
4:GOSUB80:PRINT"──────────":A=16:GOSUB50
3540 Z=15:GOSUB80:PRINT"▒"A$(F)"▓ :"TAB(
38-LEN(AB$));AB$
3550 PRINTTAB(39-LEN(AB$));:FORI=1TOLEN(
AB$)-1:PRINT"=";:NEXT
3560 Z=20:GOSUB80:IFK=1THENPRINT"♠USDRUC
K IN ─\ !":GOTO3580
3570 IFK=0THENPRINT"♠USDRUCK IN ¯ROZENT
!"
3580 IFD=0THEN3620
3590 Z=13:S=0:GOSUB80::PRINT"▒"A$(17)"▓"
:PRINT"▒"A$(18)"▓"
3600 Z=17:S=1:GOSUB80:A=19:GOSUB50:PRINT
AB$" %"
3610 PRINTTAB(7-LEN(AB$));:FORI=1TOLEN(A
B$)+1:PRINT"=";:NEXT
3620 Z=23:S=0:GOSUB80:FORI=1TO4:PRINT"▒"
BA$(I)"▓";:NEXT
3630 Y=1:GOSUB200:IFY=4THEN70
3640 IFY=3THEN1010
3650 IFY=2THEN4500
4000 REM * AENDERUNG
```




```
4010 C=1:GOSUB10:IFDTHENGOSUB40
4020 Z=2:S=13:GOSUB80::PRINT"♠ENDERUNG :
":Z=3:GOSUB80:PRINT"‾‾‾‾‾‾‾‾‾‾‾‾‾"
4030 Z=4:S=13::FORI=9TO12:GOSUB80:PRINT"
▩"A$(I)"▩ :";A(I):Z=Z+2:NEXT
4040 Z=23:S=0:GOSUB80:PRINTMD$;
4050 Z=4:Z1=Z:Z2=10:S=13:Y=9:Y1=Y:Y2=12:
SW=2:C=1:GOSUB100
4060 IFC=0THENGOSUB10:GOTO3000
4070 Z=13:S=13:GOSUB80:PRINT"▩"A$(Y)"▩ :
";PM$:GOSUB90
4080 Z=23:S=0:GOSUB80:PRINTMC$;:Z=13:S=2
7:GOSUB300:A(Y)=VAL(IN$):GOTO4010
4500 REM * PRINTER
4505 TA$=CHR$(10):TB$=CHR$(16)
4510 OPEN4,4,7:PRINT#4,CHR$(14);TB$"25♠
"
4520 PRINT#4,TB$"25------------------"TA$
4530 FORI=9TO12:PRINT#4,CHR$(15)TB$"10**
* "A$(I)" : "A(I):NEXT
4540 A=16:GOSUB50:PRINT#4,TA$;TB$"14"A$(
F)" : "AB$
4550 PRINT#4,TB$"15";:FORI=1TO2+LEN(A$(F
)+AB$):PRINT#4,"=";:NEXT
4560 PRINT#4,CHR$(13):IFDTHENGOSUB4620
4565 PRINT#4,TA$;TA$;TA$
4570 PRINT#4,CHR$(13);TB$"10***  ╯APITAL
 UND ♠INSSATZ WERDEN";
4580 PRINT#4," IN ╯ ─╲ ╯ AUSGEDRUCKT,"
4590 PRINT#4,TB$"16♠INSSATZ UND ╲AHRESZI
NS (FALLS VORHANDEN) IN ╯ % ╯ !"
4600 PRINT#4:FORI=1TO40:PRINT#4,"==";:NE
XT:PRINT#4,TA$;TA$;TA$
4610 CLOSE4:FORI=56176TO56215:POKEI,0:NE
XT:GOTO3620
4620 IFDTHENPRINT#4,TA$;TB$"15"A$(17)" "
A$(18)" : ";
4630 A=19:GOSUB50:PRINT#4,AB$
4640 PRINT#4,TB$"15";:FORI=1TOLEN(A$(18)
+A$(19)+AB$)+10:PRINT#4,"=";:NEXT
4650 RETURN
5000 REM * VARIABLEN
5010 MA$="▩▩  ♠ ╲ ╱ ♥ ‾ ♥ ♠ ╲ ╱ ♥ _ ‾ ─
▩"
5020 MB$="▩ ─3 U. ─5 ─ ○AHL ! ╯─┐ ╭─╱╯ ─
 ‾INGABE !▩"
5030 MC$="▩  ─1 ─ ╱EU !          │EBEN ♥IE E
IN !        ▩"
5040 MD$="▩ ─1 ─ ─ESULTAT !       ─3 U. ─5
 ─ ○AHL ! ▩"
5050 KO$=" ♠N.KAPITAL":P$="  ♠INSSATZ ":
KN$=" ‾NDKAPITAL"
5060 N$="  ╲AHRE    ":A$(12)=" ♠INSPER./
A":PM$=".........."
5100 BA$(1)="  ♠ENDERN "
5110 BA$(2)="  ╯OPIE   "
5120 BA$(3)="  ╱EU     "
5130 BA$(4)=" ╲ENUE    "
5500 A$(1)=KN$
5510 A$(2)=KO$
5520 A$(3)=P$
5530 A$(4)="  │AUFZEIT "
5540 A$(5)="  ╲ENUE    "
5600 A$(13)="  ♠ENDERN  "
5610 A$(14)="  ─ESULTAT "
5620 A$(15)="  ╲ENUE    "
5630 A$(17)="‾FFEKTIVER"
5640 A$(18)="╲AHRESZINS":RETURN

READY.
```

```
8 POKE45,PEEK(174):POKE46,PEEK(175):CLR
9 CLR:DIMA$(15),A(15):GOTO1000
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
20 FORZ=Z1TOZ2:GOSUB90:PRINT"
                  ":NEXT:IFC=0THENRETURN
30 Z1=12:Z2=20:C=0:GOTO20
50 RETURN
70 A$(12)=STR$(INT(A(12)*100+.5)):REM *
RUNDEN
75 AB$=LEFT$(A$(12),LEN(A$(12))-2)+"."+R
IGHT$(A$(12),2):RETURN
80 LOAD"MENUE",8:REM *LOAD MENUE
90 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION
95 A=55869:FORI=1597TO1622:POKEI,100:POK
EA,1:A=A+1:NEXT:RETURN
100 REM * AUSWAHLROUTINE *
110 GOSUB90:PRINT"▩              ▩":FORI=1T
O75:NEXTI
120 GOSUB90:PRINT"▩"A$(Y)"▩":FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDCTHENC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB90:PRINT"▩>▩":
RETURN
160 IFG$=CHR$(135)THEN190
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE *:S=0
210 Z=23:GOSUB90:PRINT"▩              ▩";:FO
RI=1TO75:NEXT
220 GOSUB90:PRINT"▩"BA$(Y)"▩";:FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB90:PRINT"▩>▩";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+10:Y=Y+1:IFS>31THENS=0:Y=1
270 GOTO210
280 S=S-10:Y=Y-1:IFS<0THENS=30:Y=4
290 GOTO210
300 REM * EINGABEROUTINE *
310 GOSUB90
320 IN$=""
330 PRINT"▄█";
340 GETG$:IFG$=""THEN340
350 G=ASC(G$):IFG=13THEN460
360 IFG=13THENRETURN
370 IFG$<"0"ORG$>"9"THEN420
380 IN$=IN$+G$
390 PRINTG$;
400 IFLEN(IN$)>NTHENFORI=1TOLEN(IN$):PRI
NTCHR$(20);:NEXT:GOTO320
410 GOTO330
420 IFG=46THEN380
425 IFG$=CHR$(133)THEN9
430 IFG<>20THEN330
440 IFLEN(IN$)<1THEN330
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO390
460 PN$="":FORI=1TO(10-LEN(IN$)):PN$=PN$
+".":NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
1000 REM ** DIKONTIEREN **
1010 GOSUB5000:PRINTMA$
```

Listing »Zinseszinsrechnung«

```
1020 A=55388:FORI=1076TO1906STEP40:POKEI
,101:POKEA,1:A=A+40:NEXT
1030 A=55736:FORI=1464TO1474:POKEI,100:P
OKEA,1:A=A+1:NEXT
1040 A=55749:FORI=1477TO1502:POKEI,100:P
OKEA,1:A=A+1:NEXT
1050 Z=23:S=0:GOSUB90:PRINTMB$
1060 Z=3:GOSUB90:PRINT"♣USWAHL :":PRINT"
────────"
1070 Z=5:FORI=1TO5:GOSUB90:PRINT"▧"A$(I)
"▪":Z=Z+1:NEXT
1080 Z=5:Z1=Z:Z2=9:S=0:Y=1:Y1=Y:Y2=5:SW=
1:C=0:GOSUB100:F=Y:IFF=5THEN80
1150 IFF=1THENA$(9)=KN$:A$(10)=P$:A$(11)
=JA$:K=1
1160 IFF=2THENA$(9)=KB$:A$(10)=P$:A$(11)
=JA$:K=1
1170 IFF=3THENA$(9)=KN$:A$(10)=KB$:A$(11
)=JA$:K=0
1180 IFF=4THENA$(9)=KN$:A$(10)=KB$:A$(11
)=P$:K=2
2000 REM * EINGABEMASKE *
2010 Z=3:S=13:GOSUB90:PRINT"¯INGABE :":Z
=4:GOSUB90:PRINT"────────────"
2020 Z=5:FORI=9TO11:S=13:GOSUB90:PRINT"▧
"A$(I)"▪ : ";PM$
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB90:PRINTMC
$
2040 Z=5:S=27:N=10:GOSUB300:A(9)=VAL(IN$
)
2050 Z=7:GOSUB300:A(10)=VAL(IN$)
2060 Z=9:GOSUB300:A(11)=VAL(IN$)
2070 Z=13:S=13:GOSUB90:PRINT"¯INGABE KOR
REKT ? (J/N):":GOSUB95
2080 Z=13:S=38:GOSUB90:PRINT"▄▓";
2090 GETG$:IFG$<>"J"ANDG$<>"N"THEN2090
2100 PRINTG$:IFG$="J"THENY=1
2110 IFG$="N"THENY=0
2120 GETI$:IFI$<>CHR$(13)ANDI$<>CHR$(20)
THEN2120
2130 IFI$=CHR$(20)THEN2080
2140 C=1:GOSUB10:IFY=0THEN2010
3000 REM * RECHNUNG *
3010 IFF=1THENA(12)=A(9)*(1/(1+(A(10)/10
0))^A(11))
3020 IFF=2THENA(12)=A(9)/(1/(1+(A(10)/10
0))^A(11))
3030 IFF=3THENA(12)=((A(9)/A(10))^(1/A(1
1))-1)*100
3040 IFF=4THENA(12)=LOG(A(9)/A(10))/LOG(
1+(A(11)/100))
3500 REM * BILDSCHIRM
3510 Z=3:S=13:GOSUB90:PRINT"¯INGABE :":Z
=4:GOSUB90:PRINT"────────────"
3520 Z=5:FORI=9TO11:GOSUB90:PRINTA$(I)"
:"A(I):Z=Z+2:NEXT
3530 Z=13:GOSUB90:PRINT"_ESULTAT :":Z=14
:GOSUB90:PRINT"────────────":GOSUB70
3540 Z=16:GOSUB90:PRINT"▧"A$(F)"▪ :";:PR
INTTAB(38-LEN(AB$));AB$
3550 PRINTTAB(39-LEN(AB$));:FORI=1TOLEN(
AB$)-1:PRINT"=";:NEXT
3560 Z=20:GOSUB90:IFK=1THENPRINT"♣USDRUC
K IN ─⟍ !":GOTO3580
3570 IFK=0THENPRINT"♣USDRUCK IN ¯ROZENT
!"
3580 Z=23:S=0:GOSUB90:FORI=1TO4:PRINT"▧"
BA$(I)"▪";:NEXT
3590 Y=1:GOSUB200:IFY=4THEN80
3600 IFY=3THEN1010
3610 IFY=2THEN4500
4000 REM * AENDERUNG *
4010 C=1:GOSUB10
4020 Z=3:S=13:GOSUB90:PRINT"♣ENDERUNG :"
:Z=4:GOSUB90:PRINT"────────────"
4030 Z=5:S=13:FORI=9TO11:GOSUB90:PRINT"▧
"A$(I)"▪ :";A(I):Z=Z+2:NEXT
4040 Z=23:S=0:GOSUB90:PRINTMD$;
4050 Z=5:Z1=Z:Z2=9:S=13:Y=9:Y1=Y:Y2=11:S
W=2:C=1:GOSUB100
4060 IFC=0THENGOSUB10:GOTO3000
4070 Z=13:S=13:GOSUB90:PRINT"▧"A$(Y)"▪ :
"PM$:GOSUB95
4080 Z=23:S=0:GOSUB90:PRINTMC$;:Z=13:S=2
7:GOSUB300:A(Y)=VAL(IN$):GOTO4010
4500 REM * AUSDRUCK *
4505 TA$=CHR$(10):TB$=CHR$(16)
4510 OPEN4,4,7:PRINT#4,CHR$(14);TB$"10─⟍
─⟍╱─♥ ╯♣╗⟍| ♣L_♥"
4520 PRINT#4,TB$"10--------------------
------"TA$
4530 FORI=9TO11:PRINT#4,CHR$(15);TB$"10*
** "A$(I)" : ";A(I):NEXT
4540 GOSUB70:PRINT#4,TA$;TB$"14"A$(F)" :
"AB$
4550 PRINT#4,TB$"15";:FORI=1TO2+LEN(A$(F
)+AB$):PRINT#4,"=";:NEXT
4560 PRINT#4,TA$;TA$;TA$
4570 PRINT#4,CHR$(13);TB$"10***    ¯NDKAP
ITAL UND |ARWERT WERDEN";
4580 PRINT#4," IN ' ─⟍ ' AUSGEDRUCKT,"
4590 PRINT#4,TB$"16DER ♦INSSATZ IN ' % '
!"
4600 PRINT#4:FORI=1TO40:PRINT#4,"==";:NE
XT:PRINT#4,TA$;TA$;TA$;TA$
4610 CLOSE4:FORI=56176TO56215:POKEI,0:NE
XT:GOTO3580
5000 REM * VARIABLENLISTE *
5010 MA$="▧▧        ─⟍♥╯╔╱|⟍¯─╱ ¯⟍╱─♥ ╯♣╗
▪"
5020 MB$="▧ _3 U. _5 - ⊙AHL ! '─╗ ⌐─╱' -
¯INGABE !▪"
5030 MC$="▧ _1 - ╱EU !        |EBEN ♥IE E
IN !      ▪"
5040 MD$="▧ _1 - _ESULTAT !      _3 U. _5
- ⊙AHL ! ▪"
5050 KN$=" ¯NDKAPITAL":P$="  ♦INSSATZ ":
JA$="  ⟍AHRE    ":PM$=".........."
5060 KB$="  |ARWERT  "
5110 BA$(1)="  ♣ENDERN "
5120 BA$(2)="  ╯OPIE   "
5130 BA$(3)="  ╱EU     "
5140 BA$(4)=" ⟍ENUE    "
5510 A$(1)="  |ARWERT  "
5520 A$(2)=" ¯NDKAPITAL"
5530 A$(3)="  ♦INSSATZ "
5540 A$(4)="  LAUFZEIT "
5550 A$(5)="  ⟍ENUE    "
5590 A$(13)="  ♣ENDERN  "
5600 A$(14)="  _ESULTAT "
5610 A$(15)="  ⟍ENUE    ":RETURN
READY.
```

Listing »Diskontieren eines Kapitals«

```
7 POKE45,PEEK(174):POKE46,PEEK(175):CLR
8 REM (C) BY KLAUS KLOEKER
9 CLR:DIMA$(15),E(50,6),E$(50,6):GOTO100
0
10 Z1=3:Z2=9:S=13:REM * MASKE LOESCHEN *
15 FORZ=Z1TOZ2:GOSUB90:PRINT"
                  ":NEXT:IFC=0THENRETURN
```

```
20 Z1=13:Z2=21:C=0:GOTO15
30 FORJ=2TO6:E$(I,J)=STR$(INT(E(I,J)*10+
.5)):REM * FORMAT TABELLE
35 E$(I,J)=LEFT$(E$(I,J),LEN(E$(I,J))-1)
+"."+RIGHT$(E$(I,J),1)
40 L$="":FORA=1TO10-LEN(E$(I,J)):L$=L$+"
.":NEXT
45 E$(I,J)=L$+E$(I,J)
50 IFLEN(E$(I,J))>10THENE$(I,J)=RIGHT$(E
$(I,J),10)
55 NEXT:RETURN
70 LOAD"MENUE",8:REM *LOAD MENUE
80 A=55909:FORI=1637TO1662:POKEI,100:POK
EA,1:A=A+1:NEXT:RETURN
90 POKE214,Z:POKE211,S:SYS58732:RETURN:R
EM * CURSORPOSITION *
95 FORI=56216TO56255:POKEI,0:NEXT:RETURN
100 REM * AUSWAHLROUTINE *
110 GOSUB90:PRINT"░                 ▓":FORI=1T
O75:NEXTI
120 GOSUB90:PRINT"░"A$(Y)"▓":FORI=1TO125
:NEXT
130 GETG$:IFG$<>CHR$(133)ANDG$<>CHR$(134
)ANDG$<>CHR$(135)ANDG$<>CHR$(13)THEN110
140 IFG$=CHR$(133)ANDCTHENC=0:RETURN
150 IFG$=CHR$(13)THENGOSUB90:A$(Y)=">"+M
ID$(A$(Y),2,10):PRINT"░"A$(Y)"▓":RETURN
155 IFG$=CHR$(134)THEN170
160 IFG$=CHR$(135)THEN190
165 GOTO110
170 Z=Z+SW:Y=Y+1:IFZ>Z2THENZ=Z1:Y=Y1
180 GOTO110
190 Z=Z-SW:Y=Y-1:IFZ<Z1THENZ=Z2:Y=Y2
195 GOTO110
200 REM * WAHL-FUSSZEILE *:S=0
210 Z=23:GOSUB90:PRINT"░                ▓";:FORI
=1TO75:NEXT
220 GOSUB90:PRINT"░"BA$(Y)"▓";:FORI=1TO1
75:NEXT
230 GETG$:IFG$<>CHR$(134)ANDG$<>CHR$(135
)ANDG$<>CHR$(13)THEN210
240 IFG$=CHR$(13)THENGOSUB90:PRINT"░>▓";
:RETURN
250 IFG$=CHR$(135)THEN280
260 S=S+8:Y=Y+1:IFS>32THENS=0:Y=1
270 GOTO210
280 S=S-8:Y=Y-1:IFS<0THENS=32:Y=5
290 GOTO210
300 REM * EINGABEROUTINE *
310 GOSUB90
320 IN$=""
330 PRINT"▄▌";
340 GETG$:IFG$=""THEN340
350 G=ASC(G$):IFG=13THENGOTO460
360 IFG=13THENRETURN
370 IFG$<"0"ORG$>"9"THEN420
380 IN$=IN$+G$
390 PRINTG$;
400 IFLEN(IN$)>NTHENFORI=1TOLEN(IN$):PRI
NTCHR$(20);:NEXT:GOTO320
410 GOTO330
420 IFG=46THEN380
425 IFG$=CHR$(133)THEN9
430 IFG<>20THEN330
440 IFLEN(IN$)<1THEN330
450 IN$=LEFT$(IN$,LEN(IN$)-1):GOTO390
460 PN$="":FORI=1TO(9-LEN(IN$)):PN$=PN$+
".":NEXT:PRINTPN$
470 IFLEN(IN$)<1THEN310
480 RETURN
```

Listing »Diskontieren eines Kapitals« (Fortsetzung)

```
900 REM * MASKE *
910 PRINTMA$:A=55388:FORI=1076TO1906STEP
40:POKEI,101:POKEA,1:A=A+40:NEXT
920 A=55776:FORI=1504TO1514:POKEI,100:PO
KEA,1:A=A+1:NEXT
930 A=55785:FORI=1517TO1546:POKEI,100:PO
KEA,1:A=A+1:NEXT
950 Z=3:S=0:GOSUB90:PRINT"♠USWAHL :":PRI
NT"──────────":PRINT"░ - - - - - ▓"
960 Z=6:FORI=1TO3:GOSUB90:PRINT"░"A$(I)"
▓":PRINT"░ - - - - - ▓":Z=Z+2:NEXT
970 RETURN
999 :
1000 REM ** TILGUNGSRECHNUNG *
1010 GOSUB6000:GOSUB900
1020 Z=23:S=0:GOSUB90:PRINTMB$
1080 Z=6:Z1=Z:Z2=10:S=0:Y=1:Y1=Y:Y2=3:SW
=2:C=0:GOSUB100:F=Y:IFF=3THEN70
2000 REM * EINGABEMASKE *
2010 Z=3:S=13:GOSUB90:PRINT"‾INGABE :":Z
=4:GOSUB90:PRINT"──────────"
2020 Z=5:FORI=4TO6:S=13:GOSUB90:PRINT"░"
A$(I)"▓ : ";PM$
2030 Z=Z+2:NEXT:Z=23:S=0:GOSUB90:PRINTMC
$
2040 Z=5:S=27:N=9:FORJ=4TO6:GOSUB300:A(J
)=VAL(IN$):Z=Z+2:NEXT
2070 Z=14:S=13:GOSUB90:PRINT"‾INGABE KOR
REKT ? (J/N):":GOSUB80
2080 Z=14:S=38:GOSUB90:PRINT"▄▌";
2090 GETG$:IFG$<>"J"ANDG$<>"N"THEN2090
2100 PRINTG$:IFG$="J"THENY=1
2110 IFG$="N"THENY=0
2120 GETI$:IFI$<>CHR$(13)ANDI$<>CHR$(20)
THEN2120
2130 IFI$=CHR$(20)THEN2080
2140 C=1:GOSUB10:IFY=0THEN2010
3000 REM * RECHNUNG *
3005 PRINT"▙":Z=23:S=23:GOSUB90:PRINTMG$
:EE(3)=0:EE(5)=0
3010 IFF=2THEN3500
3100 REM * RATENTILGUNG *
3110 T1=A(4)/A(6):J2=0:T2=0
3120 FORI=1TOA(6):R1=A(4)-J2*T2:Z1=R1*(A
(5)/100):A1=Z1+T1
3130 E(I,2)=R1:E(I,3)=Z1:E(I,4)=T1:E(I,5
)=A1:E(I,6)=A1/12:GOSUB30
3140 FORJ=3TO5STEP2:EE(J)=EE(J)+E(I,J):N
EXT
3150 J2=J2+1:T2=(A(4)/A(6)):NEXT:GOTO370
0
3500 REM * AUNNUITAETENTILGUNG *
3510 Q1=A(5)/100+1:T1=A(4)*((Q1-1)/(Q1^A
(6)-1))
3520 Z1=A(4)*(A(5)/100):A1=T1+Z1:R1=A(4)
3530 E(1,2)=R1:E(1,3)=Z1:E(1,4)=T1:E(1,5
)=A1:E(1,6)=A1/12:I=1:GOSUB30
3540 T2=T1:FORI=2TOA(6):R1=R1-T2:Z2=R1*(
A(5)/100):T2=T2*Q1
3550 E(I,2)=R1:E(I,3)=Z2:E(I,4)=T2:E(I,5
)=A1:E(I,6)=A1/12:GOSUB30
3560 FORJ=3TO5STEP2:EE(J)=EE(J)+E(I,J):N
EXTJ:NEXTI
3570 EE(3)=EE(3)+E(1,3):EE(5)=EE(5)+E(1,
5)
3700 EE(3)=EE(3)/2:EE(5)=EE(5)/2
3710 FORI=3TO5STEP2:EE$(I)=STR$(INT(EE(I
)*10/.5))
3720 EE$(I)=LEFT$(EE$(I),LEN(EE$(I))-1)+
"."+RIGHT$(EE$(I),1):NEXT
```

```
3800 REM * SCREEN *
3805 GOSUB900
3810 Z=3:S=13:GOSUB90:PRINT"¯INGABE :":Z
=4:GOSUB90:PRINT"────────"
3820 Z=5:S=13:FORI=4TO6:GOSUB90:PRINTA$(
I)" :"A(I):Z=Z+2:NEXT
3830 Z=14:S=13:GOSUB90:PRINT"_ESULTAT :"
:Z=15:GOSUB90:PRINT"─────────"
3840 Z=17:FORJ=3TO5STEP2:GOSUB90:PRINT"▒
"EC$(J)"▓:"EE$(J):Z=Z+2:NEXT
3850 Z=21:GOSUB90:PRINT"♠USDRUCK IN '─\'
 !"
3860 ZA=VAL(EE$(3))/(VAL(EE$(5))/100)
3870 Z=14:S=0:GOSUB90:PRINT"▒   ♣INS-
▓":Z=15:GOSUB90:PRINT"▒   ANTEIL  ▓"
3880 ZA$=STR$(ZA):IFZA<10THENZA$=" "+ZA$
3890 ZA$=LEFT$(ZA$,5):Z=17:S=2:GOSUB90:P
RINTZA$;" %"
3900 Z=18:S=3:GOSUB90:PRINT"======"
3910 Z=23:S=0:GOSUB90:FORI=1TO5:PRINT"▒"
BA$(I)"▓";:NEXT
3920 Y=1:GOSUB200:IFY=5THEN70
3930 IFY=4THEN9
3940 IFY=3THEN5500
3950 IFY=2THEN5000
4000 REM * TABELLE *
4010 GOSUB95:X1=2:X2=4:Y1=1:Y2=A(6)
4020 Z=1:S=0:GOSUB90:FORI=1TO22:PRINTMD$
:NEXT:Z=22:GOSUB90:PRINTMH$
4030 Z=3:S=0:GOSUB90:PRINTM$(1)
4040 Z=3:S=6:FORI=X1TOX2:GOSUB90:PRINTM$
(I):S=S+11:NEXT
4050 Z=4:S=1:GOSUB90:FORI=1TO37:PRINT"="
;:NEXT
4100 Z=6:S=1:FORI=Y1TOY2
4105 I$=STR$(I):IFI<10THENI$="."+I$
4107 GOSUB90:PRINT"."+I$:S=6
4110 FORJ=X1TOX2
4120 GOSUB90:PRINTE$(I,J)
4130 S=S+11:NEXTJ
4140 IFPEEK(214)>15THENY2=I:Y1=Y2-9:GOTO
4200
4150 Z=Z+1:S=1:NEXTI
4200 Z=17:S=1:GOSUB90:FORJ=1TO38:PRINT"-
";:NEXT
4500 WAIT203,63:GETG$
4510 IFG$=CHR$(29)THEN4550
4520 IFG$=CHR$(157)THEN4580
4530 IFG$=CHR$(17)THEN4610
4540 IFG$=CHR$(145)THEN4640
4543 IFG$=CHR$(133)THEN3800
4545 GOTO4500
4550 X1=X1+1:X2=X2+1:REM * ->
4560 IFX2>6THENX2=6:X1=4:GOTO4500
4570 GOTO4030
4580 X1=X1-1:X2=X2-1:REM * <-
4590 IFX1<2THENX1=2:X2=4:GOTO4500
4600 GOTO4030
4610 IFA(6)<10THENGOTO4500
4615 Y1=Y1+10:Y2=Y2+10:REM * DOWN
4620 IFY2>A(6)THENY2=A(6):Y1=Y2-9:IFI=A(
6)THEN4500
4630 GOTO4030
4640 IFA(6)<10THENGOTO4500
4645 Y1=Y1-10:Y2=Y2-10:REM * UP
4650 IFY1<1THENY1=1:Y2=Y1+9:IFI<11THEN45
00
4660 GOTO4030
5000 REM * AENDERUNG *
5010 C=1:GOSUB10:S=0:FORZ=13TO20:GOSUB90
:PRINT"                   ":NEXT
5020 Z=3:S=13:GOSUB90:PRINT"♠ENDERUNG :"
:Z=4:GOSUB90:PRINT"──────────"
5030 Z=5:S=13:FORI=4TO6:GOSUB90:PRINT"▒"
A$(I)"▓ :";A(I):Z=Z+2:NEXT
5040 Z=23:S=0:GOSUB90:PRINTMF$;
5050 Z=5:Z1=Z:Z2=9:S=13:Y=4:Y1=Y:Y2=6:SW
=2:C=1:GOSUB100
5060 IFC=0GOTO3000
5070 Z=14:S=13:GOSUB90:PRINT"▒"A$(Y)"▓ :
";PM$:GOSUB80
5080 Z=23:S=0:GOSUB90:PRINTMC$;:Z=14:S=2
7:GOSUB300:A(Y)=VAL(IN$)
5090 A$(Y)=" "+MID$(A$(Y),2,10):GOTO5010
5500 REM * PRINTER
5505 TB$=CHR$(16):TA$=CHR$(10)
5510 OPEN4,4,7:PRINT#4,CHR$(14)TB$"10♠US
DRUCK EINES |ILGUNGSPLANES"
5520 PRINT#4,TB$;"10---------------------
----------"TA$
5530 FORI=4TO6:PRINT#4,CHR$(15)TB$"10***
 "A$(I)" : "A(I):NEXT
5540 PRINT#4,TA$;TA$;TA$
5550 FORI=1TO6:PRINT#4,"   "M$(I);:NEXT:
PRINT#4," "
5560 FORI=1TO40:PRINT#4,"--";:NEXT:PRINT
#4
5570 FORI=1TOA(6):I$=STR$(I):IFI<10THENI
$="."+I$
5575 PRINT#4,"     ."+I$" ";
5580 FORJ=2TO6:PRINT#4," .."E$(I,J);:NEX
TJ:PRINT#4," ":NEXTI
5590 FORI=1TO40:PRINT#4,"--";:NEXT:PRINT
#4
5595 PRINT#4," |ESAMT:";
5600 PO=27:FORJ=3TO5STEP2:PRINT#4,SPC(PO
-LEN(EE$(J)))EE$(J);:PO=26:NEXT
5610 PRINT#4,TA$;TA$;TB$"10*** ♠USDRUCK
DER |ABELLE ERFOLGT IN '─\' !"
5620 PRINT#4,TB$"10*** |ERECHNUNG DER ♣I
NSEN ERFOLGT AM ¯NDE DES \AHRES !"
5630 PRINT#4,TA$;TB$"10*** |ILGUNG KONST
ANT   = _ATENTILGUNG !!"
5640 PRINT#4,TB$"10*** ♠NNUITAET KONSTAN
T = ♠NNUITAETENTILGUNG !!"
5650 PRINT#4:FORI=1TO40:PRINT#4,"==";:NE
XTI:PRINT#4,TA$;TA$;TA$;TA$
5660 CLOSE4:GOTO3860
6000 REM * VARIABLEN *
6010 MA$="▒▒    | ╮ L |  ╱ ╱ | ♥ _ ¯ ─ |
   ▓"
6020 MB$="▒ _3 U. _5 - ▯AHL ! '─┐ ─/' -
 ─INGABE !"
6030 MC$="▒ _1 - ♠USWAHL !   |EBEN ♥IE E
IN !      ▓"
6040 MD$="
       "
6050 ME$="▒
       ▓"
6060 MF$="▒ _1 - _ESULTAT !      _3 U. _5
 - ▯AHL ! ▓"
6070 MG$="|ITTE ▯ARTEN !!!"
6080 MH$="▒ _1 - _ESULTAT !     ─URSOR -
♥TEUERN ! ▓"
6100 A$(1)=" _ATENTILG."
6110 A$(2)=" ♠NNU.TILG."
6120 A$(3)=" \ENUE     "
6130 A$(4)=" ─ARLEHEN  "
6140 A$(5)=" ♣INSSATZ  "
6150 A$(6)=" \AHRE     "
```

Listing »Tilgungsplan«

```
6160 PM$="........."
6200 M$(1)=" ╰AHR "
6210 M$(2)="_ESTSCHULD"
6220 M$(3)="     ♦INSEN"
6230 M$(4)="    |ILGUNG"
6240 M$(5)=" ♠NNUITAET"
6250 M$(6)=" ♠NNU./╲ON"
6300 EC$(3)=" ♦INSEN  GES."
6310 EC$(5)=" ♠NNUIT. GES."
6400 BA$(1)=" |ABELLE"
6410 BA$(2)=" ♠ENDERN"
6420 BA$(3)="  ╯OPIE "
6430 BA$(4)="   ╱EU  "
6440 BA$(5)="  ╲ENUE "
6500 RETURN
READY.
```

Listing »Tilgungsplan« (Schluß)

Fortsetzung von Seite 67

HYPRA-LOAD: Schnelles Laden von Diskette

Warum Zeit verschwenden? Laden Sie Ihre Programme in Zukunft fünfmal schneller.

Schon ganze Scharen von Computerbesitzern werden hilflos gestöhnt haben, als sie wieder einmal eine Kaffeepause einlegen mußten, um das Laden des nächsten Programms von Diskette abzuwarten. War das Programm dann auch noch länger als 150 Blocks, so konnte diese »Zwangssendepause« durchaus länger als 2,5 Minuten werden.

Auch Hardware-Hersteller haben diesen Mangel erkannt und bieten neuerdings immer mehr schnelle Laufwerke als Ersatz für die VC 1541 an.

Halt! Jetzt aber Schluß mit dem dauernden Herziehen über die VC 1541, die nämlich eigentlich gar keine Schuld an den langen Wartezeiten hat; vielmehr kann sie es, was die Geschwindigkeit betrifft, ohne weiteres mit den großen CBM-Floppys aufnehmen, oder anders formuliert, die VC 1541 ist genauso schnell wie zum Beispiel die CBM 2031.

Aber wo liegt denn dann nun der Haken? Das Stichwort lautet: Der serielle Bus, also die Schnittstelle zwischen Floppy und Computer. Diese Verbindung erfordert zwar weniger Materialaufwand als der üblicherweise bei Commodore benutzte IEEE-488-Bus, (der 8-Bit-parallel arbeitet). Der serielle Bus hat aber den Nachteil, daß die Geschwindigkeit bei der Übertragung im Normalbetrieb zirka 5 bis 6mal langsamer ist als beim IEC-Bus, was sich dann auch auf die Ladezeit auswirkt.

Die Hardwarelösung habe ich ja schon erwähnt: Sie erfordert außer einer neuen Floppy, auch noch das IEEE-Bus-Modul, da der C 64 in der Grundversion keine derartige Schnittstelle zu bieten hat. Mitunter ist also diese Lösung sehr teuer und übersteigt die Kaufkraft vieler C 64-Anwender.

Aber zum Glück gibt es noch eine Softwaremöglichkeit, hier effektiv einzugreifen. Das Prinzip ist denkbar einfach. Es beruht auf der Möglichkeit, in die Floppy Maschinenprogramme einzuspeichern und diese dann dort auszuführen.

In unserem Fall werden also modifizierte Busroutinen in der Floppy und im Computer abgelegt, die dann die sonst so langsame Übertragung auf ein Vielfaches steigern.

In Zahlen: Der serielle Bus wird durch diese Routinen weit über 10mal so schnell und das Laden geschieht nun mit der maximalen Floppygeschwindigkeit, die etwa 5 bis 6mal höher ist.

Nun aber zum Programm, damit Sie so schnell wie möglich in den Genuß von HYPRA-LOAD kommen: Wenn Sie den Basic-Lader eintippen, achten Sie bitte besonders sorgfältig auf richtiges Eingeben der Checksummen, da diese zur Fehlerkontrolle bei der Eingabe der DATAs dienen!

Vor dem ersten Startversuch das Programm bitte unbedingt auf Diskette speichern, da es sich bei der Ausführung automatisch überschreibt. Anschließend starten Sie mit RUN. Wurde alles korrekt eingegeben, wird am Basic-Anfang ein Maschinenprogramm generiert und das Basic-Programm gelöscht.

Dieses Maschinenprogramm ist das eigentliche HYPRA-LOAD und kann nun mit »SAVE« abgespeichert werden.

Um den Geschwindigkeitsvorteil auch wirklich nutzen zu können, wird empfohlen, HYPRA-LOAD möglichst auf alle Programmdisketten zu überspielen. Es wurde dazu extra sehr kurz (6 Blocks) gehalten. Müßten Sie nämlich nach jedem Laden zuerst die Diskette wechseln, so wäre die Effektivität von HYPRA-LOAD erheblich vermindert.

Übrigens: Die Einschaltmeldung von HYPRA-LOAD umfaßt auch eine Bytes-free-Meldung, welche jedoch ignoriert werden sollte, da die Zahlenangabe aus programmtechnischen Gründen falsch ist. Dies hat jedoch keinen Einfluß auf die einwandfreie Funktion des Computers!!!

Nun aber zu den besonderen Eigenschaften — HYPRA-LOAD kopiert nach dem Kaltstart das gesamte Betriebssystem vom ROM ins darunterliegende RAM, jedoch ohne dabei Basic-Speicherplatz zu verbrauchen. Danach werden die einzelnen Programmteile im Betriebssystem verteilt und das Betriebssystem im RAM gestartet.

— Abzulesen ist die Aktivierung von HYPRA-LOAD an der Speicherstelle 1 der Zeropage. Diese enthält normalerweise den Wert 55, was besagt, daß ab $A000 ROM aktiviert ist, welches Basic und das Betriebssystem enthält.

HYPRA-LOAD schaltet den Wert auf 53 um und aktiviert dadurch anstelle von ROM-Bausteinen freien RAM Bereich, in dem nun das Betriebssystem und Basic stehen.

Man kann also ganz einfach HYPRA-LOAD abschalten, durch POKE 1,55
und, sofern nicht zerstört, mit
POKE 1,53
wieder einschalten.

— Da HYPRA-LOAD im RAM steht, ergeben sich natürlich einige Probleme. So gibt es viele Programme, die beim Laden den Bereich ab $A000 überschreiben. Würde nach der Rückkehr aus der Laderoutine nun ins RAM ab $A000 gesprungen, so wäre ein »Aufhängen« des Computers fast unvermeidlich.

Gegen solche Programme ist HYPRA-LOAD abgesichert, und es wird nach dem Ladevorgang automatisch ins ROM-Basic zurückgesprungen. Bei mehrteiligen Programmen erkennt man dies daran, daß plötzlich wieder »normal« geladen wird.

Überschreiben Programme jedoch den Bereich ab $E000, so treffen sie früher oder später auf die Laderoutinen und der Computer »stürzt ab«. Bei solchen Programmen hilft nur das vorherige Ausprobieren. Es muß an dieser Stelle erwähnt werden, daß es auch Programme gibt, die ein Abschalten von HYPRA-LOAD erfordern, aber erst nachdem der Ladevorgang abgeschlossen ist. Auch hier hilft zum Erkennen solcher Programme nur das Ausprobieren.

Am besten arbeitet HYPRA-LOAD mit einteiligen Programmen, da es hier nicht vorkommt, daß der Bereich ab $E000 überschrieben wird (solche Programme müßten nämlich den davorliegenden I/O-Bereich überschreiben) und so das gesamte Programm schnell geladen werden kann!

— Es sollte vielleicht noch einmal erwähnt werden, daß sich HYPRA-LOAD nur auf den LOAD-Befehl beschränkt, andere Diskettenoperationen bleiben davon unberührt.

— Da HYPRA-LOAD im Betriebssystem steht, vernichtet es zwangsläufig andere Routinen, das heißt, vor Kassetten- und RS232-Betrieb muß HYPRA-LOAD abgeschaltet werden.

— SYS 64738 als Reset reaktiviert auch HYPRA-LOAD, sofern es noch nicht zerstört oder ausgeschaltet ist.

— Nach einem Reset durch einen Taster kann HYPRA-LOAD, sofern noch nicht überschrieben, durch

POKE 1,53

wieder eingeschaltet werden. Wird danach jedoch langsam geladen, so ist im Basic-Teil etwas verändert worden, so daß HYPRA-LOAD auf das ROM umgeschaltet hat.

— Ein Test hat ergeben, daß HYPRA-LOAD auf dem SX 64 trotz dessen angeblicher »Vollkompatibilität« nicht lauffähig ist.

— Da zusätzliche Peripheriegeräte und auch der Bildschirm den schnellen Busbetrieb stören, müssen diese entfernt, beziehungsweise abgestellt und der Bildschirm ausgeschaltet werden. Ist dies nicht geschehen, so macht HYPRA-LOAD darauf aufmerksam und wartet auf das Nachholen des Versäumten. Anschließend kann durch Drücken der RUN/STOP-Taste mit dem Laden begonnen werden. Es erfolgt keine weitere Kontrolle mehr!

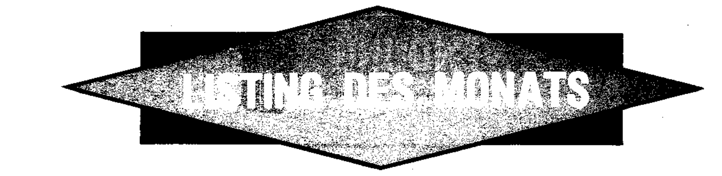
— Betreiben Sie ein Floppy-Laufwerk zum Beispiel unter der Geräteadresse 9, so müssen folgende Änderungen durchgeführt werden:

Die DATAs in den Zeilen 104, 114 und 239; dort müssen die entsprechenden Zahlen (008) auf die neue Geräteadresse umgestellt werden. Natürlich ist auch ein Angleichen der Checksummen erforderlich. Es sind dies die 1., die 2. und die 12. Checksumme.

Übrigens: Für eventuelle Berichte über die Erfahrungen mit HYPRA-LOAD sowie Verbesserungsvorschläge wäre ich sehr dankbar.

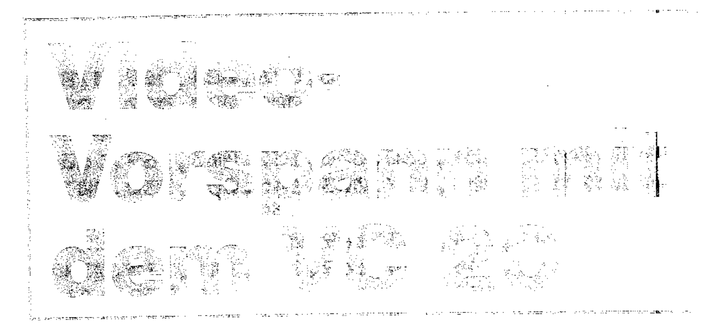
Und nun viel Spaß mit einer 5 bis 6mal so schnellen VC 1541. (K. Schramm/B. Schneider/gk)

```
0 :CLR:GOTO380
10 ************************************
11 ***      ==================       ***
12 ***      HYPRA-LOAD-SYSTEM        ***
13 ***      ==================       ***
14 ***                               ***
15 ***      WRITTEN 1984 BY :        ***
16 ***      """"""""""""""""""       ***
17 ***                               ***
18 ***     - KARSTEN SCHRAMM -       ***
19 ***     - BORIS SCHNEIDER -       ***
20 ***                               ***
21 ************************************
100 DATA011,008,192,007,158,051,050,051
101 DATA056,000,000,000,032,047,243,169
102 DATA226,162,248,133,167,134,168,169
103 DATA000,162,003,133,169,134,170,169
104 DATA008,032,012,237,169,111,032,185
105 DATA237,169,077,032,221,237,169,045
106 DATA032,221,237,169,087,032,221,237
107 DATA160,000,165,169,032,221,237,165
108 DATA170,032,221,237,169,030,032,221
109 DATA237,177,167,032,221,237,200,192
110 DATA030,144,246,032,254,237,024,165
111 DATA167,105,030,133,167,144,003,230
112 DATA168,024,165,169,166,170,105,030
113 DATA133,169,144,002,230,170,224,005
114 DATA144,173,201,000,144,169,169,008
115 DATA032,012,237,169,111,032,185,237
116 DATA169,077,032,221,237,169,045,032
117 DATA221,237,169,069,032,221,237,169
118 DATA139,032,221,237,169,004,032,221
119 DATA237,169,011,141,017,208,032,251
120 DATA237,234,234,076,046,240,169,011
121 DATA141,000,221,044,000,221,016,251
122 DATA169,003,141,000,221,162,005,202
123 DATA234,208,252,162,004,173,000,221
124 DATA042,042,102,176,106,102,176,234
125 DATA202,208,242,165,176,073,255,096
126 DATA032,082,239,201,255,240,248,160
127 DATA000,169,011,141,000,221,044,000
128 DATA221,016,251,169,003,141,000,221
129 DATA162,007,202,208,253,173,000,221
130 DATA042,042,102,176,106,102,176,234
131 DATA234,173,000,221,042,042,102,176
132 DATA106,102,176,234,234,173,000,221
133 DATA042,042,102,176,106,102,176,234
134 DATA234,173,000,221,042,042,102,176
135 DATA106,102,176,165,176,073,255,153
136 DATA166,251,200,208,180,096,120,169
137 DATA001,133,167,160,255,032,124,239
138 DATA192,255,240,064,162,002,165,167
139 DATA240,002,162,004,173,166,251,208
140 DATA007,238,167,251,173,167,251,044
141 DATA169,000,133,168,189,166,251,145
142 DATA174,230,174,208,002,230,175,232
143 DATA228,168,208,240,162,000,134,167
144 DATA173,166,251,208,198,169,053,133
145 DATA001,169,027,141,017,208,169,064
146 DATA133,144,024,096,169,053,133,001
147 DATA169,027,141,017,208,234,169,027
148 DATA056,096,160,000,185,066,240,153
149 DATA048,001,200,192,031,208,245,032
150 DATA210,239,076,048,001,234,008,072
151 DATA169,000,168,089,000,160,200,208
152 DATA250,201,128,240,004,169,055,133
153 DATA001,166,174,164,175,104,040,088
154 DATA096,221,016,173,240,034,173,161
155 DATA002,074,176,250,173,001,221,041
156 DATA253,141,001,221,173,001,221,041
157 DATA004,240,249,169,144,024,076,059
158 DATA239,173,161,002,041,018,240,243
159 DATA024,096,173,151,002,172,156,002
160 DATA204,155,002,240,011,041,247,141
161 DATA151,002,177,247,238,156,002,096
162 DATA009,008,141,151,002,169,000,096
163 DATA072,173,161,002,240,017,173,161
164 DATA002,041,003,208,249,169,016,141
165 DATA013,221,169,000,141,161,002,104
166 DATA096,165,000,041,006,201,002,240
167 DATA003,076,158,253,234,169,005,133
168 DATA009,162,090,134,075,162,000,169
169 DATA082,133,036,032,086,245,080,254
170 DATA184,173,001,028,197,036,240,009
171 DATA198,075,208,239,169,010,076,105
172 DATA249,080,254,184,173,001,028,149
173 DATA037,232,224,007,208,243,032,151
174 DATA244,165,022,069,023,069,024,069
```




```
175 DATA025,069,026,240,007,198,009,208
176 DATA192,076,030,244,165,024,197,006
177 DATA240,003,076,011,244,133,034,169
178 DATA006,133,049,076,060,004,165,018
179 DATA166,019,133,022,134,023,165,006
180 DATA133,024,165,007,133,025,169,000
181 DATA069,022,069,023,069,024,069,025
182 DATA133,026,032,052,249,162,090,032
183 DATA086,245,160,000,080,254,184,173
184 DATA001,028,217,036,000,240,006,202
185 DATA208,237,076,081,245,200,192,008
186 DATA208,234,032,086,245,080,254,184
187 DATA173,001,028,145,048,200,208,245
188 DATA160,186,080,254,184,173,001,028
189 DATA153,000,001,200,208,244,032,224
190 DATA248,165,056,197,071,240,003,076
191 DATA246,244,032,233,245,197,058,240
192 DATA003,076,002,245,160,000,169,085
193 DATA032,082,004,185,000,006,133,119
194 DATA044,000,024,016,251,169,016,141
195 DATA000,024,044,000,024,048,251,162
196 DATA000,138,102,119,042,042,102,119
197 DATA042,042,141,000,024,138,102,119
198 DATA042,042,102,119,042,042,141,000
199 DATA024,138,102,119,042,042,102,119
200 DATA042,042,141,000,024,138,102,119
201 DATA042,042,102,119,042,042,141,000
202 DATA024,162,002,202,208,253,169,015
203 DATA141,000,024,200,208,173,234,234
204 DATA234,234,234,234,234,173,000,028
205 DATA009,008,141,000,028,173,000,006
206 DATA208,003,076,158,253,197,024,208
207 DATA249,133,006,173,001,006,133,007
208 DATA076,101,003,133,119,044,000,024
209 DATA016,251,169,016,141,000,024,044
210 DATA000,024,048,251,162,004,169,000
211 DATA102,119,042,042,102,119,042,042
212 DATA141,000,024,202,208,240,234,234
213 DATA234,234,234,234,169,015,141,000
214 DATA024,096,096,133,000,088,165,000
215 DATA048,252,120,096,120,234,234,234
216 DATA234,234,234,165,024,141,000,006
217 DATA133,006,165,025,141,001,006,133
218 DATA007,169,004,133,120,169,226,032
219 DATA130,004,201,002,144,051,160,000
220 DATA132,120,164,120,185,219,254,240
221 DATA018,088,032,118,214,120,169,226
222 DATA032,130,004,201,002,144,026,230
223 DATA120,208,231,169,192,032,130,004
224 DATA169,226,032,130,004,201,002,144
225 DATA008,169,255,032,082,004,076,034
226 DATA235,173,000,006,240,248,197,024
227 DATA240,196,173,000,006,133,006,173
228 DATA001,006,133,007,076,160,004,234
229 DATA234,234,234,160,000,185,025,244
230 DATA153,048,001,200,192,031,208,245
231 DATA076,048,001,169,000,168,089,000
232 DATA160,200,208,250,201,128,240,007
233 DATA169,055,133,001,076,001,245,076
234 DATA048,244,160,000,177,187,201,036
235 DATA240,242,169,001,133,167,169,000
236 DATA133,144,165,167,032,012,237,169
237 DATA111,032,185,237,165,144,016,011
238 DATA230,167,165,167,201,016,208,230
239 DATA076,187,238,165,167,201,008,240
240 DATA239,160,000,185,122,244,240,006
241 DATA032,210,255,200,208,245,032,225
242 DATA255,208,251,238,234,234,234,234
243 DATA076,187,238,234,013,066,073,084
244 DATA084,069,032,078,085,082,032,070
245 DATA076,079,080,080,089,032,065,078
246 DATA083,067,072,065,076,084,069,078
247 DATA013,000,000,000,000,000,000,000
248 DATA000,000,000,000,000,120,169,055
249 DATA133,001,160,000,132,003,169,160
250 DATA133,004,177,003,145,003,230,003
251 DATA208,248,230,004,208,244,169,229
252 DATA141,214,253,169,076,141,114,254
253 DATA141,249,244,169,188,141,115,254
254 DATA169,254,141,116,254,169,009,141
255 DATA250,244,169,244,141,251,244,169
256 DATA018,133,003,169,010,133,004,169
257 DATA226,133,005,169,248,133,006,177
258 DATA003,145,005,200,208,249,230,004
259 DATA230,006,165,006,201,250,208,239
260 DATA169,016,162,008,133,003,134,004
261 DATA169,187,162,238,133,005,134,006
262 DATA177,003,145,005,200,208,249,230
263 DATA004,230,006,165,006,201,240,208
264 DATA239,185,067,013,153,095,228,200
265 DATA192,080,208,245,160,000,185,012
266 DATA012,153,009,244,200,192,148,208
267 DATA245,169,053,133,001,076,248,252
268 DATA234,234,000,032,066,089,084,069
269 DATA083,032,070,082,069,069,013,000
270 DATA000,000,000,000,000,000,147,013
271 DATA032,032,032,032,042,042,042,042
272 DATA032,067,054,052,045,072,089,080
273 DATA082,065,045,076,079,065,068,045
274 DATA083,089,083,084,069,077,032,042
275 DATA042,042,042,013,013,032,040,067
276 DATA041,056,052,032,018,084,082,073
277 DATA066,065,082,146,032,032,000,129
278 DATA072,032,000,000,000,000,000,000
279 DATA000,000,000,000,000,000,000,000
280 REM ******************************
290 REM *         CHECKSUMMEN          *
300 REM ******************************
310 DATA10355,10666,10986,11346,11058
320 DATA09936,08856,08690,09460,07556
330 DATA08117,08952,09829,10579,08282
340 DATA12151,09771,03416,00000,00000
350 REM ******************************
360 REM *     INITIALISIERE DATAS      *
370 REM ******************************
380 PRINT"[illegible]  HABEN SIE MICH ABGE SAVED?"
390 GETA$:IFA$<>"J"ANDA$<>"N"THEN390
400 IFA$="N"THENPRINT:PRINT:PRINT"DANN A
BER SCHNELL [illegible]":END
410 PRINT:PRINT:PRINT"BITTE GEDULD; UEBE
RPRUEFE CHECKSUMMEN !"
420 DIM S(18)
430 FORX=1TO18
440 FORY=1TO80:READA:S(X)=S(X)+A
450 NEXTY,X
460 FORX=1TO18
470 READCS:IFS(X)=CSTHENNEXT:GOTO510
480 PRINT:PRINT:PRINT"FEHLER IN DATENBLO
CK !!!"
490 PRINT:PRINT"ZEILEN";90+10*X;" -";99+
10*X" !"
500 END
510 RESTORE:PRINT:PRINT:PRINT"DATEN WERD
EN ABGESPEICHERT ."
520 FORX=0TO1439:READA:POKE2049+X,A:NEXT
530 POKE45,179:POKE46,13:POKE47,179:POKE
48,13
540 PRINT:PRINT"MIT 'SAVE' SPEICHERN ":P
RINT
550 PRINT"MIT 'RUN' AKTIVIEREN !"
560 CLR
READY.
```

Listing. Basic-Lader von »HYPRA-LOAD«



Einen professionellen Vorspann wie aus einem richtigen Filmstudio produziert dieses Programm.

Erzeugt wird ein technischer Vorspann, ähnlich wie ihn auch Fernsehstudios verwenden. Dieser Vorspann dient einerseits der Kontrolle der richtigen Einstellungen, andererseits der Dokumentation der einzelnen Szenen. Darüber hinaus wird die Synchronisation einer eventuellen Nachvertonung erleichtert.

Der Vorspann besteht im einzelnen aus
— einem farbfolgerichtigen Farbbalken zur Einstellkontrolle; dazu wird ein Meßton von 1000 Hz erzeugt,
— einem »Etikett« (anstatt der Klappe beim Film) mit Angaben über Datum, Bandnummer, Geschwindigkeit, Aufnahmegerät, Quelle, Serienname und Titel,
— schließlich einem Countdown von neun bis zwei im Sekundentakt, dazu wieder ein Meßton von 1000 Hz.

Unter dem Farbbalken und über den Zahlen des Countdowns erscheint Ihr Name beziehungsweise die Studiobezeichnung. Ersetzen Sie daher in den Zeilen 440 und 1330 »Schulz Video« durch die von Ihnen gewünschte Bezeichnung.

Nach dem Starten des Programms muß als erstes die Eingabe der Etikett-Daten erfolgen. Folgende Angaben sind notwendig:
— Bandnummer (bis 9 Stellen)
— Geschwindigkeit (SP oder LP)
— Datum (beliebiges Format)
— Aufnahmegerät (bis 13 Stellen)
— Quelle (bis 13 Stellen)
— Serienname (bis 19 Stellen)
— Titel (bis 19 Stellen)

BAND-NR. 64 SP
09.08.84
GERAET: EVERTSBISHI F
QUELLE: SUMPF
SERIE:
REDAKTION IN PANIK
TITEL:
WO IST MEIN ARTIKEL

Jede Antwort wird mit RETURN abgeschlossen. Die Fortschaltung der Sequenzen Farbbalken/Etikett/Countdown geschieht mit der F7-Taste. Zwischen den einzelnen Sequenzen erscheint immer — bis zum erneuten Drücken von F7 — eine Schwarzblende.

Nach dem Countdown bewirkt die F7-Taste wiederum das Erscheinen der Farbbalken, und die einzelnen Sequenzen werden erneut durchlaufen. Für eine neue Beschriftung des Etiketts ist STOP/RESTORE und ein erneuter Programmstart notwendig.

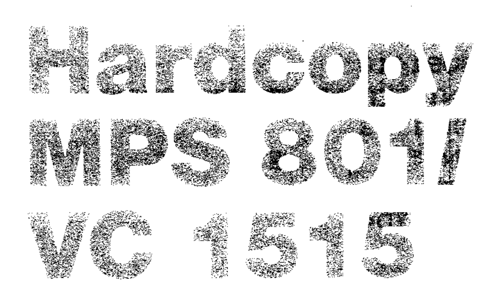
(Wolfgang Schulz/ev)

```
1 REM*****************
2 REM*VIDEO-VORSPANN*
3 REM*VON           *
4 REM*WOLFGANG R.   *
5 REM*SCHULZ        *
6 REM*THEODOR-      *
7 REM*KOERNER-WEG 5 *
8 REM*2000 HAMBURG61*
9 REM****************
10 POKE36879,8:PRINT"■"
20 PRINT"■BANDNUMMER EINGEBEN■"
30 PRINT:PRINT:PRINT:PRINT
40 INPUTBN$:PRINT"■■■GESCHWINDIGKEIT SP
/LP"
50 PRINT"■■":INPUT GS$
60 GOSUB1360
70 PRINT:PRINT:PRINT"■■AUFZEICHNUNGSGERA
ET"
80 PRINT:PRINT:PRINT
90 INPUT G$
100 PRINT:PRINT:PRINT"■■SERIEN-NAME■"
110 PRINT:PRINT:PRINT
120 INPUTSE$
130 PRINT:PRINT:PRINT"■TITEL■"
140 PRINT:PRINT:PRINT: INPUTT$
150 PRINT:PRINT:PRINT"■QUELLE■"
160 PRINT:PRINT
170 INPUTQ$
180 GOSUB1490:GOSUB1280:PRINT"■":GOSUB12
80
190 GOSUB 1300
200 PRINT"■":GOSUB1280
210 BN$=LEFT$(BN$,6):GS$=LEFT$(GS$,2)
220 PRINT"■BAND-NR. ■"BN$;:PRINTTAB(19)"
"GS$
230 PRINT"■■┌──────────────────┐"
240 PRINT"|                  |"
250 PRINT"└──────────────────┘"
260 PRINT" ■■■■ "DT$
270 G$=LEFT$(G$,13)
280 PRINT"■■■■GERAET: ■"G$
290 Q$=LEFT$(Q$,13)
300 PRINT:PRINT"■QUELLE:■ "Q$
310 PRINT:PRINT
320 PRINT"■■■━━━━━━━━━━━━━━━━━■ "
330 FORS=1TO7:PRINT"▌
▌"
340 NEXTS
350 PRINT"■■━━━━━━━━━━━━━━━━━■ "
360 PRINT"■■■■■■■■■■■SERIE:■"
```




```
370 SE$=LEFT$(SE$,19)
380 PRINT"■■"SE$
390 PRINT"■■■TITEL:■"
400 T$=LEFT$(T$,19)
410 PRINT"■■"T$
420 GOSUB1280
430 PRINT"■":GOSUB1280
440 PRINT"■■     SCHULZ    VIDEO"
450 PRINT"■      ________________"
460 PRINT"■":PRINT
470 FORI=1TO3
480 PRINT"          ■        "
490 NEXTI
500 FORM=1TO4
510 PRINT"          ■ ■■■■■  "
520 NEXT M
530 FORZ=1TO3
540 PRINT"          ■        "
550 NEXTZ
560 FORB=1TO7
570 PRINT"          ■■■■■■■■ "
580 NEXTB
590 GOSUB1410
600 PRINT"■■■■■■■■■"
610 FORP=1TO4
620 PRINT"          ■ ■■■■■  "
630 NEXTP
640 FORO=1TO3
650 PRINT"          ■        "
660 NEXTO
670 GOSUB1410
680 PRINT"■■■■■■■■■■■■■■■■■"
690 FORU=1TO4
700 PRINT"               "
710 NEXTU
720 FORK=1TO10
730 PRINT"                 "
740 NEXTK
750 PRINT"■■■■■■■■■■■■■■■■■■■■■"
760 GOSUB1410
770 FORB=1TO3
780 PRINT"          ■      "
790 NEXTB
800 FORL=1TO4
810 PRINT"          ■ ■        "
820 NEXTL
830 FORJ=1TO3
840 PRINT"          ■        "
850 NEXTJ
860 FORX=1TO4
870 PRINT"          ■ "
880 NEXTX
890 FORZ=1TO3
900 PRINT"          ■      "
910 NEXTZ
920 GOSUB1410
930 PRINT"■■■■■■■■■"
940 FORV=1TO4
950 PRINT"               "
960 NEXTV
970 FORJ=1TO55:NEXTJ
980 GOSUB1410
990 PRINT"■■■■■■■■■■■■■■■■■"
1000 FORA=1TO7
1010 PRINT"          ■ ■    ■  "
1020 NEXTA
1030 PRINT"■■■■■■■■"
1040 FORT=1TO3
1050 PRINT "                 "
1060 NEXTT
1070 GOSUB1410
1080 PRINT"■■■■■■■■■■■■■■■■■■■■■"
1090 FORX=1TO3
1100 PRINT"          ■       "
1110 NEXTX
1120 FORQ=1TO4
1130 PRINT"                ■ "
1140 NEXTQ
1150 PRINT"■■■■■■■■"
1160 FORJ=1TO3
1170 PRINT"          ■       ":NEXTJ
1180 GOSUB1410
1190 POKE36876,0
1200 PRINT"■■■■■■■■■"
1210 FORK=1TO4
1220 PRINT"          ■ ■■■■■    "
1230 NEXTK
1240 FORJ=1TO300:NEXT
1250 POKE36878,0
1260 PRINT"■":GOSUB1280
1270 GOTO1300
1280 GETF$:IFF$<>"■"THENGOTO1280
1290 RETURN
1300 FORJ=1TO19:POKE36876,239:POKE36878,
7
1310 PRINT"■■  ■  ■  ■  ■  ■  ■
"
1320 NEXTJ
1330 PRINT:PRINT"■    SCHULZ    VIDEO"
1340 GOSUB1280:PRINT"■":POKE36876,0
1350 GOTO200
1360 PRINT:PRINT:PRINT"■■DATUM"
1370 PRINT:PRINT"■":INPUTDT$
1380 PRINT:PRINT"■"
1390 RETURN
1400 END
1410 REM MESSTON
1420 FORPIEP=1TO15
1430 POKE36878,7
1440 POKE36876,239
1450 NEXTPIEP
1460 POKE36878,1
1470 FORN=1TO625:NEXT
1480 RETURN
1490 PRINT"■"
1500 PRINT"■STARTE JEDES ELEMENT"
1510 PRINT:PRINT"MIT DER (F-7)-TASTE."
1520 PRINT:PRINT
1530 PRINT"■DIE SEQUENZEN SIND:"
1540 PRINT:PRINT"■1. SCHWARZ/STUMM"
1550 PRINT:PRINT"2. FARBBALKEN/MESSTON"
1560 PRINT:PRINT"3. SCHWARZ/STUMM"
1570 PRINT:PRINT"4. ETIKETT"
1580 PRINT:PRINT"5. SCHWARZ/STUMM"
1590 PRINT:PRINT"6. COUNTDOWN"
1600 PRINT:PRINT"7. SCHWARZ/STUMM"
1610 RETURN
READY.
```

(Schluß)



Mit dem Programm kann in knapp drei Minuten der Grafikbildschirm des C 64 ausgedruckt werden. Dabei wird Bit für Bit der Inhalt des Grafikspeichers auf die Druckernadel übertragen. Es arbeitet mit dem MPS 801 und läuft auch mit dem VC 1515. Allerdings muß dann der Bildschirm während des Druckens abgeschaltet werden.

Der Basic-Lader kann auch in eigenen Grafik-Programmen verwendet werden. Sie können den Bereich 8192 bis 49151 als Grafikspeicher benutzen.

Um passende Bits für jede Druckernadel im Grafikspeicher zu finden, habe ich den Ausdruck BYTEADRESSE = 8192 + 8 * X + Y + INT(Y/8) * 312 verwendet. Er läßt sich aus den Angaben im »Commodore 64-Programmers Reference Guide« (Seiten 124 bis 126) ableiten. X und Y sind dabei die Koordinaten eines Grafikpunktes (gezählt von der oberen, linken Ecke des Bildschirms). Die Gleichung gibt zu jedem Bildpunkt die Adresse des Bytes, in dem das entsprechende Bit gesetzt (Bildpunkt an) oder gelöscht (Bildpunkt aus) ist.

Diese Adressen müssen für den MPS 801-Drucker in Siebener-Blöcke sortiert werden, da der Druckkopf nur über sieben Nadeln verfügt. Jeder Siebener-Block steuert dann achtmal (einmal für jedes Bit) die sieben Druckernadeln.

Insgesamt ist der Grafikbildschirm 320 Punkte entsprechend — 320 : 8 = 40 Bytes — breit. Eine Druckzeile enthält also 40 * 7 = 280 Bytes. Das Programm sortiert jeweils 280 Bytes (1 Druckzeile) in Siebener-Blöcke und speichert sie im Bereich 49920 bis 50199. Von dort werden die Bitmuster auf den Druckkopf übertragen.

Das Sortieren erfolgt mit der Formel POKE(49920+7*X+Y-INT(Y/7)*7),PEEK(BYTEADRESSE).

Mit Hilfe der Additions-, Multiplikations- und Divisionsroutinen aus R. Zaks :»Programmierung des 6502« und L.A. Leventhal :»6502 Programmieren in Assembler« habe ich diesen Ausdruck in Maschinensprache übersetzt.

Benutzung des Programms

CHR$(8) schaltet den Grafik-Modus des Druckers ein. Die drei folgenden CHR$(13) bewirken einen Papiervorschub. Er sorgt für ein sauberes Drucken der ersten Grafik-Zeilen (Zeile 450). Aufgerufen wird die Druckroutine mit SYS 49525,LF (Zeile 460). Dabei ist LF die logische Filenummer des geöffneten Drucker-Files (Zeile 450). Mit CHR$(15) wird der Drucker wieder auf den Normal-Modus eingestellt (Zeile 470).

In den Zeilen 410 bis 430 wird die Anfangsadresse des Grafikspeichers ermittelt und in Zeile 440 der Hardcopy-Routine mitgeteilt. Dadurch läßt sich zum Beispiel auch der RAM-Bereich unter dem Basic-Interpreter als Grafikspeicher nutzen. Der ganze Basic-Bereich (2048 bis 40959) ist dann für Programme frei.

Nicht benutzt werden darf das RAM unter dem Betriebssystem, da es zur Ausgabe der Zeichen auf den Drucker gebraucht wird (BSOUT=$FFD2). Ebenfalls nicht benutzbar als Grafikspeicher ist der Bereich 49152 bis 57343, da sonst das Programm von der Grafik überschrieben wird.

Die Zeilen 410 bis 440 können weggelassen werden, wenn der Grafikspeicher bei 8192 bis 16191 liegt.

(Rainer Kracht/rg)

Basic-Lader »Hardcopy MPS 801/VC 1515«

```
320 REM ***********************
330 REM *  HARDCOPY-LOADER    *
340 REM *                     *
350 REM *  RAINER KRACHT      *
360 REM *  GAERTNERTWIETE 9   *
370 REM *  2085 QUICKBORN     *
380 REM *                     *
390 REM *  MAERZ 1984         *
400 REM ***********************
410 :
420 :
430 DATA 0,32,253,174,32,158,183,32,201,
255,162,29,142,116,193,169,0,133,251
440 DATA 133,255,169,8,32,210,255,32,233
,193,169,0,160,195,133,176,132,177
450 DATA 169,13,32,210,255,169,40,133,21
,169,128,133,151,169,0,133,20,160
460 DATA 6,177,176,37,151,240,7,165,20,2
5,226,193,133,20,136,16,240,165,20
470 DATA 9,128,32,210,255,70,151,144,223
,165,176,105,6,133,176,144,2,230,177
480 DATA 198,21,208,205,206,116,193,208,
180,169,13,32,210,255,76,204,255,1
490 DATA 2,4,8,16,32,64,160,39,132,252,1
69,0,133,248,162,8,10,38,248,6,252
500 DATA 144,7,24,105,7,144,2,230,248,20
2,208,239,133,247,24,165,255,101,247
510 DATA 133,247,169,195,101,248,133,248
,169,0,133,250,152,201,32,48,2,230
520 DATA 250,10,10,10,133,249,24,165,251
,101,249,133,249,169,0,101,250,133
530 DATA 250,24,169,32,101,250,133,250,1
65,251,74,74,74,133,252,133,253,169
540 DATA 0,133,254,162,8,10,38,254,6,253
,144,7,24,105,56,144,2,230,254,202
550 DATA 208,239,133,253,24,165,252,101,
254,133,254,24,165,253,101,249,133
560 DATA 249,165,254,101,250,133,250,162
,0,165,1,133,2,169,54,133,1,161,249
570 DATA 129,247,165,2,133,1,136,48,3,76
,235,193,165,251,201,199,240,18,230
580 DATA 251,230,255,165,255,201,7,240,3
,76,233,193,169,0,133,255,96,169,195
590 DATA 133,248,169,4,133,247,162,40,16
9,0,160,2,145,247,136,16,251,165,247
600 DATA 24,105,7,133,247,144,2,230,248,
202,208,233,96
610 FORI=49524TO49854:READA:POKEI,A:B=B+
A:NEXT
620 IFB<>45611THENPRINT"FEHLER IN DATENZ
EILE !"
READY.
```

Der VC 1526/MPS 802 als Grafikdrucker

Aufgabe des Programms ist es, die Punkte aus dem Bildspeicher (nachfolgend als HRBM — Hires-Bit-Map — bezeichnet) zu lesen und dem Drucker in geeigneter Form zu übergeben. Die gestellte Aufgabe läßt sich in die folgenden Abschnitte unterteilen: Initialisierung; Lesen des HRBM; Umformen der gelesenen Daten; Übergabe der Daten an den Drucker; Ende.

Betrachten wir das Ganze aus der Bit-Sicht. Das HRBM besteht aus 8 000 Bytes. Diese lassen sich als 8 x 8 Bit-Zeichen zusammenfassen. Man erhält dann also 25 Zeilen zu je 40 Spalten aus 8 x 8 Bit-Zeichen = 1000 solche Zeichen. Das Programm liest also ein solches Zeichen aus der HRBM und definiert nach diesem Muster einen neuen Character für den 1526. Im HRBM sind aber die Bytes horizontal orientiert, während sie der Drucker vertikal orientiert erwartet (siehe Skizze 1). Das Umformungsprogramm hat deshalb die Aufgabe, die Bits um die Diagonale zu spiegeln. Damit wären alle Teilprobleme gelöst.

Das Assemblerprogramm bietet noch folgende Vorzüge:

— Befehlssyntax: SYS 6 * 4096, T, P
  T = Tabulator (0< =T<=40)
  P*4096 = Anfangsadresse des HRBM (0< =P< =15)
— das HR-Bild kann auf dem Drucker in der Horizontalen verschoben werden (Tab)
— durch einen speziellen Parameter wird das HRBM gewählt (auch das »versteckte« RAM unter dem Interpreter und dem Betriebssystem)
— durch RUN/STOP unterbrechbar (Files werden korrekt geschlossen)
— kein Zeilenvorschub, das heißt zwei nacheinander ausgedruckte Bilder fügen sich nahtlos aneinander
— das Programm liegt von $6 000 bis $613b in einem günstigen Bereich und kann somit mit vorhandenen Basic-Erweiterungen verwendet werden
— hohe Geschwindigkeit durch reine Maschinensprache.

Wenn Sie das Hardcopy-Programm in Verbindung mit einer Basic-Erweiterung verwenden, beachten Sie bitte folgende Reihenfolge:
1. Basic-Erweiterung laden und starten.
2. Basic-Lader des Hardcopy-Programms laden und starten.
Das Maschinenprogramm schützt sich selbst vor dem Überschreiben.
Es kann sein, daß Sie vor dem Starten des Programms den Zeilenvorschub des VC 1526/MPS 802 verkleinern müssen:
OPEN 4,4: OPEN 6, 4, 6: PRINT#6, CHR$(18): PRINT#4:
CLOSE 4: CLOSE 6

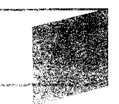
(Lucas Kalt/rg)

Tabelle 1.
Dies sind die Angaben, die das Programm benötigt.

Simons Basic	P=14
Screen-Graphics 64	P=10
Ultra-Basic 64	P=10
Supergrafik 64	P=10 & P=14
Hi-Res Grafik Aid (64 intern)	P=2

Basic-Lader »Hardcopy VC 1526/MPS 802«

```
1 REM HI RES HARDCOPY V10
2 REM DURCH SYS 6*4096,T,P WIRD EIN HI R
ES HARD-
3 REM COPY  AUF  DEN  DRUCKER  1526  AUS
GEGEBEN.
4 REM T             =    TABULATOR   (0<=
T <= 40)
5 REM P * 4096      =    ANFANGSADRESSE  D
ES  HRBM
6 REM Z.B. SYS 6*4096,20,2
7 REM
8 REM V1  29.12.83 DO BY LK
9 REM V10 25. 1.84 MI BY LK
10 REM
11 REM      '84 BY LUCAS KALT
12 REM
13 REM NICHT RELOKATIBEL!!!
14 FORA=6*4096TO6*4096+1*256+3*16+11:REA
DB:X=X+B:POKEA,B:NEXT
15 IFX<>40377THENPRINT"** DATA-FEHLER **
":STOP
16 PRINT"SYS 6*4096,T,P":PRINT
17 PRINT"T       = TABULATOR (0<=T<=40)
18 PRINT"P*4096= ANFANGSADRESSE DES HRBM
19 POKE56,6*16:POKE55,1:NEW
20 DATA169,106,162,4,160,6,32,212,96,162
,106,32,201,255,169,20,32,210,255
21 DATA169,105,162,4,160,5,32,212,96,169
,100,162,4,160,0,32,212,96,32,253
22 DATA174,32,158,183,224,41,176,12,134,
252,32,253,174,32,158,183,224,16
23 DATA144,6,32,179,96,76,72,178,138,10,
10,10,10,133,255,162,0,134,254,142
24 DATA60,3,169,0,141,61,3,32,223,96,32,
28,97,240,41,162,105,32,201,255
25 DATA162,0,189,62,3,32,210,255,232,224
,8,208,245,169,13,32,210,255,162
26 DATA100,32,201,255,32,42,97,169,254,3
2,210,255,169,141,32,210,255,24
27 DATA165,254,105,8,133,254,144,2,230,2
55,32,225,255,240,30,238,61,3,173
28 DATA61,3,201,40,208,181,162,100,32,20
1,255,169,13,32,210,255,238,60,3
29 DATA173,60,3,201,25,208,156,169,105,3
2,195,255,169,13,32,210,255,169
30 DATA100,32,195,255,162,106,32,201,255
,169,36,32,210,255,32,204,255,169
31 DATA106,76,195,255,32,186,255,169,0,3
2,189,255,76,192,255,120,165,1,41
32 DATA253,133,1,162,0,138,157,62,3,232,
224,8,208,248,169,128,133,253,160
33 DATA0,177,254,162,0,10,144,10,72,189,
62,3,5,253,157,62,3,104,232,224
34 DATA8,208,238,70,253,200,192,8,208,22
7,165,1,9,2,133,1,88,96,162,0,138
35 DATA221,62,3,208,5,232,224,8,208,246,
96,24,165,252,109,61,3,170,240,8
36 DATA169,32,32,210,255,202,208,250,96
READY.
```



Der besondere Leckerbissen — die mehrfarbige Hardcopy

Mit Ihrem Commodore 64 und einem VC 1520-Plotter können Sie farbige Hardcopys produzieren.

Das Programm arbeitet als Basic-Lader mit Maschinencodeteil in Form von DATA-Zeilen. Der Maschinenspracheteil operiert im Bereich ab $ 6000, der meines Wissens bei allen gängigen Basic-Versionen zur Verfügung steht. Das bedeutet, daß das Programm mit ihnen lauffähig ist und Hardcopys ihrer Grafikseiten machen kann, selbst wenn diese, wie bei Simons Basic, unter dem ROM liegen. Man braucht nur die Startadresse der Grafikseite einzugeben, und schon setzt sich der Plotter in Bewegung. Die Kopierdauer liegt bei zirka 12 bis 13 Minuten für eine einfarbige Kopie und bei einer halben bis dreiviertel Stunde für eine mehrfarbige Kopie. Dabei gilt: je geringer die Bildgröße, desto schneller die Kopie. Aber auch eine große Grafik wird dann schnell ausgedruckt, wenn sie wenig unterbrochene Linien enthält. (Klaus Schneider/rg)

```
10 REM MASCHINENSPRACHEHARDCOPYROUTINE
20 REM          1520/C64
30 REM          13.6.1984
40 REM             BEI
50 REM       KLAUS SCHNEIDER
60 REM        BAHNHOFSTR. 5
70 REM         493 DETMOLD
80 REM
90 REM *********** EINGABE ************
100 REM
110 PRINT CHR$(147)
120 PRINT " SOLL DIE KOPIE MEHRFARBIG WE
RDEN (J/N)"
130 GET A$ : IF A$="" THEN 130
140 IFA$="J"THENPOKE1000,1:PRINT" JA":G=
3
150 IFA$="N"THENPOKE1000,0:PRINT" NEIN":
G=1
160 FORI=1TOG
170 PRINT" BITTE GEBEN SIE DIE":I;"■. FA
RBE (0-3) AN!"
180 GET B$ : IF B$<"0" OR B$>"3" THEN 18
0
190 POKE1000+I,VAL(B$):PRINT" "+B$
200 PRINT
210 NEXTI
220 REM
230 REM ****** TEST AUF ROUTINE *******
240 REM
250 FOR I=0 TO 10
260 READ A
270 B=PEEK(24576+I)
280 IF B<>A THEN 310
290 NEXT I
300 GOTO 440
310 RESTORE
320 REM
330 REM ****** POKEN DER ROUTINE ******
340 REM
350 PRINT CHR$(147)
360 PRINT TAB(13) "BITTE WARTEN"
370 FOR I=0 TO 825
380 READ A
390 C=C+A
400 POKE 24576+I,A
410 NEXT I
420 IF C<>87544 THEN PRINT " FEHLERHAFTE
 DATEN !!!" : END
440 REM
450 REM ***** EINGABE DER ADRESSE *****
460 REM
470 PRINT CHR$(147)
480 PRINT " BITTE GEBEN SIE DIE STARTADR
ESSE DER "
490 PRINT
500 INPUT " GRAFIK EIN! ";A
510 B=INT(A/256)
520 POKE 24647,B : POKE 24640,A-B*256
530 REM
540 REM ***** STARTEN DER ROUTINE *****
550 REM
560 SYS 6*4096
570 PRINT CHR$(147)
580 REM
590 REM ****** GGF. WIEDERHOLUNG ******
600 REM
610 PRINT " WOLLEN SIE NOCH EINE KOPIE (
J/N) ?"
620 GET A$ : IF A$="" THEN 620
630 IF A$<>"J" THEN END
640 PRINT
650 PRINT " MIT DENSELBEN PARAMETERN (J/
N) ?"
660 GET A$ : IF A$="" THEN 660
670 IF A$<>"J" THEN RUN
680 GOTO 560
690 REM
700 REM ********** DIE DATEN **********
710 REM
720 DATA 162,0,142,14,220,120,134,183,23
2,134,184,134,185,162,6,134,186,32,192
730 DATA 255,162,2,134,184,134,185,162,0
,134,183,162,6,134,186,32,192,255,162
740 DATA 0,142,64,3,142,65,3,142,67,3,14
2,70,3,142,71,3,142,72,3,142,73,3,142
750 DATA 76,3,162,0,134,247,142,148,97,1
```

```
62,224,134,248,142,152,97,162,1,142
760 DATA 75,3,162,208,142,163,96,142,215
,96,162,2,32,201,255,173,233,3,24,105
770 DATA 48,32,210,255,169,13,32,210,255
,32,204,255,169,52,133,1,160,0,177,247
780 DATA 162,55,134,1,72,24,165,247,105,
8,144,2,230,248,133,247,173,232,3,240
790 DATA 56,104,162,0,142,74,3,10,46,74,
3,10,46,74,3,72,170,174,74,3,236,75
800 DATA 3,208,3,32,0,98,238,73,3,238,65
,3,208,3,238,64,3,238,65,3,208,3,238
810 DATA 64,3,173,73,3,201,4,208,203,76,
238,96,104,162,0,142,74,3,24,10,46,74
820 DATA 3,72,173,74,3,205,75,3,208,3,32
,0,98,238,73,3,238,65,3,208,3,238,64
830 DATA 3,173,73,3,201,8,208,215,162,0,
142,73,3,104,238,72,3,173,72,3,201,40
840 DATA 176,3,76,112,96,173,76,3,240,3,
32,0,98,238,67,3,173,78,3,141,65,3,173
850 DATA 77,3,141,64,3,32,0,98,162,0,142
,72,3,142,64,3,142,65,3,142,76,3,169
860 DATA 208,141,163,96,141,215,96,198,2
48,165,247,56,233,63,176,2,198,248,133
870 DATA 247,238,71,3,173,71,3,201,8,240
,3,76,112,96,162,0,142,71,3,230,248
880 DATA 165,247,24,105,56,144,2,230,248
,133,247,238,70,3,173,70,3,201,25,240
890 DATA 3,76,112,96,32,204,255,173,232,
3,208,3,76,246,97,238,75,3,173,75,3
900 DATA 201,4,240,119,162,0,142,70,3,14
2,67,3,142,64,3,142,65,3,142,77,3,142
910 DATA 78,3,162,0,134,247,162,224,134,
248,162,1,32,201,255,162,0,138,72,189
920 DATA 231,97,32,210,255,104,170,232,2
24,9,208,241,162,1,32,201,255,162,0
930 DATA 138,72,189,240,97,32,210,255,10
4,170,232,224,6,208,241,162,2,32,201
940 DATA 255,173,75,3,56,233,1,168,185,2
33,3,24,105,48,32,210,255,169,13,32
950 DATA 210,255,32,204,255,76,112,96,77
,44,48,44,45,50,48,49,13,77,44,48,44
960 DATA 48,13,32,47,243,169,1,141,14,22
0,88,96,173,64,3,72,173,65,3,72,173
970 DATA 67,3,72,162,0,142,67,3,142,68,3
,142,69,3,201,100,144,8,233,100,238
980 DATA 67,3,76,23,98,201,10,144,8,233,
10,238,68,3,76,35,98,201,1,144,8,233
990 DATA 1,238,69,3,76,47,98,173,65,3,17
4,64,3,240,17,162,2,142,64,3,162,5,142
1000 DATA 65,3,232,142,66,3,76,95,98,162
,0,142,64,3,142,65,3,142,66,3,201,100
1010 DATA 144,8,238,64,3,233,100,76,95,9
8,201,10,144,8,238,65,3,233,10,76,107
1020 DATA 98,201,1,144,8,238,66,3,233,1,
76,119,98,173,65,3,24,105,8,141,65,3
1030 DATA 173,66,3,201,10,144,8,233,10,1
41,66,3,238,65,3,173,65,3,201,10,144
1040 DATA 11,233,10,141,65,3,238,64,3,76
,155,98,162,1,32,201,255,173,76,3,240
1050 DATA 8,169,68,32,210,255,76,196,98,
169,77,32,210,255,169,44,32,210,255,173
1060 DATA 64,3,24,105,48,32,210,255,173,
65,3,24,105,48,32,210,255,173,66,3,24
1070 DATA 105,48,32,210,255,169,44,32,21
0,255,169,45,32,210,255,173,67,3,24,105
1080 DATA 48,32,210,255,173,68,3,24,105,
48,32,210,255,173,69,3,24,105,48,32,210
1090 DATA 255,169,13,32,210,255,104,141,
67,3,104,141,65,3,104,141,64,3,173,163
1100 DATA 96,73,32,141,163,96,141,215,96
,173,76,3,73,1,141,76,3,173,65,3,141
1110 DATA 78,3,173,64,3,141,77,3,96
READY.
```

Listing »Farbige Hardcopy VC 1520«

Hardcopy Gemini-10X

Das Programm erlaubt, in der Kombination des von C 64 mit dem Matrix-Drucker Gemini-10X von Star eine Hardcopy der hochauflösenden Grafik zu erstellen.

Es werden die Grafikpunkte in 25 Zeilen zusammengefaßt und nacheinander ausgedruckt.

Der Druckvorgang dauert rund 45 Sekunden.

Das Programm wird in den Kassettenspeicher eingelesen und von dort mit folgendem Befehl aufgerufen:

xxx OPENX,4,5
xxx SYS828,X
xxx PRINT # X:CLOSEX

Das X bei den einzelnen Befehlen steht für die freidefinierbare Filenummer.

Nach dem CLOSE-Befehl springt das Programm automatisch wieder in das Basic-Programm.

(Gunnar E. Krüger/rg)

```
10 REM HARDCOPY C64 UND GEMINI-10X
15 REM MIT STAR-UNIVERSAL-INTERFACE
20 FOR I=828 TO 943
30 READ X:POKEI,X:S=S+X:NEXT I
40 DATA76,63,3,32,253,174,32,158,183,32,
201,255,169,0,160,32
50 DATA133,253,132,254,162,25,160,7,44,1
60,5,185,160,3,32
60 DATA210,255,136,16,247,169,40,133,21,
169,128,133,151,169
70 DATA0,133,20,160,7,177,253,37,151,240
,7,165,20,25,168,3
80 DATA133,20,136,16,240,165,20,32,210,2
55,70,151,144,225
90 DATA165,253,105,7,133,253,144,2,230,2
54,198,21,208,207
100 DATA202,208,189,169,13,32,210,255,76
,204,255,1,64,75,27
110 DATA13,8,65,27,128,64,32,16,8,4,2,1
120 IF S<>13426 THEN PRINT"FEHLER":STOP
130 PRINT"OK!"
READY.
```

Basic-Lader »Hardcopy Gemini-10X«

Olympia Compact 2: ein Interface

Das hier beschriebene Interface ermöglicht den Anschluß einer Olympia Compact 2 an den Commodore 64. Es gleicht einerseits die Zeichensätze der beiden Geräte an die ASCII-Norm an, andererseits gestattet es den Ausdruck der Steuerzeichen des C 64.

Die Steuerzeichen werden in einer Form dargestellt, anhand derer man sie bequem identifizieren und wieder eingeben kann. Außerdem simuliert das Programm eine Rücktaste und führt einen Cursor für den Drucker.

Für die Verbindung des Userports mit der Centronics-Schnittstelle ist ein Kabel erforderlich, das folgende Pins verbindet:

Userport	8	C	D	E	F	H	J	K	L	B	A und N
Amphenol	1	2	3	4	5	6	7	8	9	11	19 — 29

PC2 des C 64 liefert also das Strobe-Signal, das der Schreibmaschine das Vorhandensein gültiger Daten mitteilt, und an FLAG2 wird das BUSY-Signal des Druckers auf Bereitschaft zur Entgegennahme neuer Zeichen abgefragt.

Das Programm können Sie natürlich auch ohne die REMs abtippen, die hier wiedergegebene Pseudo-Assembler-Fassung hat den Sinn, einen gewissen Durchblick zu ermöglichen und Änderungen zu vereinfachen.

Die Schnittstelle hat die Sekundäradresse 4. Wird eine andere Geräteadresse gewünscht, etwa, weil schon ein Matrixdrucker mit dieser Adresse existiert, so sind die Zeilen 127 und 137 des Programms entsprechend zu ändern.

Das Maschinenprogramm belegt in der vorliegenden Fassung die Speicherplätze von 32000 bis 32511. So ist eine Zusammenarbeit mit Simons Basic und Exbasic II gewährleistet. Wer das Programm in einen anderen Bereich, zum Beispiel in die 4 KByte RAM ab 49152, legen will, muß dazu die Zeilen 79, 111, 114, 116, 142, 144, 218, 224, 233, 237 und 256, die durch ein Sternchen gekennzeichnet sind, ändern. In 111 ist das MSB der Anfangsadresse einzusetzen (bei Anfang 50000 ist dies 195), in 114 das LSB der Anfangsadresse des CHROUT-Teils (Anfang + 55, im Beispiel 135), in 116 die Anfangsadresse des CHKOUT-Teils (Anfang + 30, im Beispiel 110). In 142 und 144 stehen jeweils als zweites Byte LSB und MSB vom Anfang des Umsetzungskatalogs. Sie wären bei Umlegung der Schnittstelle nach 50000 auf 73 (Zeile 144) und 196 (Zeile 140) zu ändern.

Das Programm verwendet die Adresse »Anfang« (32000) als Druckercursor, der für formatierten Druck abgefragt werden kann (zum Beispiel mit PRINT#4 TAB(31-PEEK (32000)),...). Im Programm dient er dazu, durch (von der Compact 2 nicht tatsächlich ausgeführten, sondern druckwegoptimierten) Wagenrücklauf und Wiedervorrücken eine Rücktaste zu simulieren. Dieser Cursor wird in den Zeilen 218, 224, 233, 237 und 256 des Programms verwendet und müßte bei Verlegung des Programms geändert werden (für Anfang 50000 ist jeweils das LSB 80 statt 0 und das MSB 195 statt 125).

Zum Aufbau des Programms: Der Initialisierungsteil verbiegt die KERNAL-Vektoren CHKOUT und CHROUT in die Schnittstellensoftware. Da diese Vektoren durch RUNSTOP/ RESTORE wieder zurückgestellt werden, muß die Initialisierungsroutine nicht nur nach dem Laden, sondern nach jedem RUNSTOP/ RESTORE mit SYS 32001 aufgerufen werden. Die neu eingebauten Teile CHKOUT und CHROUT prüfen nun bei jeder Zeichenausgabe, ob das Zeichen für die Schreibmaschinenschnittstelle bestimmt ist.

Der CHROUT-Teil sucht ein auszugebendes Zeichen zunächst im Sonderzeichenkatalog. Hier wird ein Zeichen des C 64 in bis zu drei Codes an die Schreibmaschine umgesetzt. (Man beachte das zusammengestoppelte Potenzzeichen oder das elegante Pi). Dabei werden die Steuerzeichen für die Schreibmaschine im Druckmodus anders behandelt als im Listmodus. Ist das auszugebende Zeichen nicht im Sonderzeichenkatalog zu finden, so wird es je nach Zeichensatz auf ASCII-Wert umgesetzt. Anstelle von unbekannten Zeichen druckt die Schreibmaschine als Joker einen accent aigu »´«.

In den folgenden drei Abschnitten des Programms werden die angesammelten Schreibmaschinencodes der Reihe nach ausgegeben und der Cursor aktualisiert.

Zum Katalog der Sonderzeichen: Hier steht an erster Stelle der Wagenrücklauf, der beim C 64 anders als bei der Schreibmaschine gehandhabt wird. Jeder Wagenrücklauf (CR, CHR$(13)), den der C 64 sendet, muß durch eine Zeilenschaltung (LF, CHR$(10)) ergänzt werden. Das zu ersetzende Zeichen (hier 13) steht immer in der vierten Spalte des Kataloges, die Ersatzzeichen (hier 13,10,0) in den ersten drei Spalten. Das Zeichen 0 wird nur deswegen gesendet, weil jedes umzusetzende C64 Sonderzeichen prinzipiell durch drei Compact 2-Zeichen ersetzt wird.

Das folgende Sonderzeichen ermöglicht es, bei Bedarf auch nur den Wagenrücklauf zu senden. Der neue Code hierfür ist 11 (Befehl PRINT CHR$(11) oder in Strings PRINT »<Control>K«. Die dritte Umsetzung ist erforderlich, damit die Compact 2 auch ein geshiftetes Space als solches serviert bekommt.

Danach folgen 8 Steuerzeichen der Schreibmaschine, die mit PRINT CHR$(1, 2, 3, 21, 22, 23, 25 oder 26) oder in einen String auch mit PRINT »<Control>A, B, C, U, V, W, Y oder Z« gesendet werden können. Dies ist auch mit PRINT CHR$(12) beziehungsweise PRINT »<Control>L« möglich, was einen Seitenvorschub bewirkt. Die Control-Methode führt jedoch bei A,B,C und L zu Problemen, da sie nicht gelistet, sondern auch im Listmodus ausgeführt werden.

Im Katalog folgen nun die Umlaute und Sonderzeichen der Schreibmaschine, die den folgenden Tasten des C 64 zugeordnet wurden:

Comm + : ä
Shift + : Ä
Comm — : ö

Shift — : Ö
Pfund : ß
Comm Pfund : ü
Shift Pfund : Ü
Comm Alphak : §
Shift Alphak : l
Comm ★ : ´
Shift ★ : Unterstreichen des nächsten Zeichens
eckige Klammer auf : [2]
eckige Klammer zu : [3]

Der Akzent »´« wird über das nachfolgende Zeichen gedruckt, UNT+RT unterstreicht das nachfolgende Zeichen. Auf den Einbau von Promille und My wurde verzichtet. Wem diese Zeichen lieb und wert sind, der möge eine der Zeilen 298 bis 303 ändern, indem er die ersten drei DATA durch 60,0,0 für My und durch 14,46,15 für Promille ändert. (Die vierte DATA-Zahl ist, wie erwähnt, der Code des C 64-Zeichens, das ersetzt wird). In TEXT 64, das das Pfundzeichen als Steuerzeichen benutzt, könnte man sich das »ß« erhalten, indem man die 64 und die 92 in den Zeilen 288 und 294 vertauscht und so das »ß« auf die Alphakringel-Taste legt. Ich selbst habe stattdessen die Zeilen 3160, 7110, 540, 1200, 7200, 8220 und 7140 von TEXT 64 geändert.

Bei den nun anstehenden Listcodes herrscht folgende Systematik: Die Kleinbuchstabencodes gehören zu den direkt ansprechbaren Funktionen CLR, HOME und den Cursortasten. »g« bedeutet CHR$(142), Umschalten auf Großschrift, was nur mit Tricks eingegeben werden kann, aber in manchen Listings auftaucht.

Die Zeichen"!,", #, $, %, &, ' und (" stehen für die acht Farben, die zusammen mit der Commodore-Taste eingegeben werden.

Ziffern und Großbuchstaben werden zusammen mit <Control> eingegeben. Es sind dies die ersten acht Farben, 9 und 0 für RVS und RVS OFF, die Umschaltung auf den zweiten Zeichensatz durch <Control>N, Blockieren und Freigabe der Umschaltung durch <Control>H beziehungsweise <Control>I sowie die oben schon erwähnten Steuercodes <Control>F, K,U,V,W,Y,Z für die Schreibmaschine. Die angegebenen Control-Codes gehören natürlich in einen String nach PRINT.

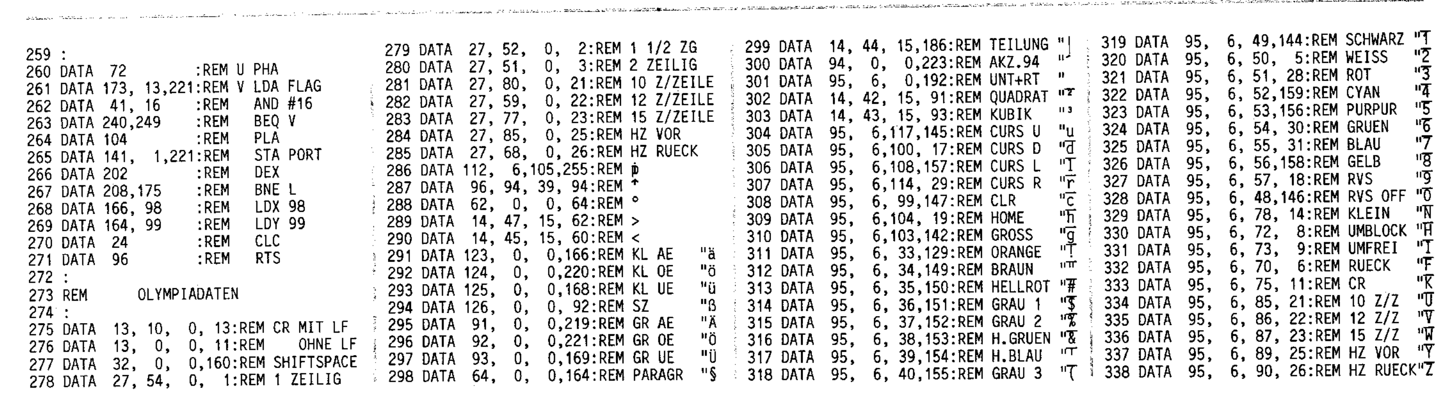
Nicht in Print-Statements, sondern nur in Abfragen tauchen mitunter auch die Listcodes der Funktionstasten auf, die aber aus Platzgründen hier ausgelassen werden mußten und von der Schreibmaschine wie alle unbekannten Zeichen als Joker »`« ausgegeben werden.

Damit nicht jedesmal das ganze, lange Programm eingelesen werden muß, habe ich am Anfang des Programms eine Möglichkeit beschrieben, eine reine Maschinencodefassung abzuspeichern und direkt oder unter Programmkontrolle wieder einzulesen.

(Reinhard Atzbach/rg)

Listing: Interface für Olympia Compact 2

```
75 REM ****** CENTRONICS ************
76 REM INTERFACE FUER OLYMPIA COMPACT 2
77 REM  (C) REINHARD ATZBACH 130784
78 REM ********************************
79 A=32000:L=510:REM * ANFANG/LAENGE *
80 FORN=0TOL:READB:POKEA+N,B:NEXT
81 SYS A+1 : END
82 REM ********************************
83 REM * MASCHINENPG SAVEN: RUN:GOTO 85
84 REM
85 OPEN5,8,5,"K-CENTRONICS,P,W"
86 PRINT#5,CHR$(A AND 255);CHR$(A/256);
87 FORN=ATOA+510:PRINT#5,CHR$(PEEK(N));
88 NEXTN : CLOSE5
89 REM
90 REM * LADEN DES MASCHINENPG/DIREKT
91 REM   LOAD"K-CENTRONICS",8,1
92 REM   ENDEPOINTER 45/46 NOTIEREN !
93 REM * LADEN DES MASCHINENPG/PROGRAMM
94 REM   OPEN 5,8,5,"K-CENTRONICS,P,R"
95 REM   GET#5,A$:GET#5,A$
96 REM   FOR N=0 TO 510:GET A$
97 REM   POKE ANFANG+N,ASC(A$+CHR$(0))
98 REM   NEXT N:CLOSE 5:SYS ANFANG+1
99 REM ********************************
100 :
101 REM      DRUCKERCURSOR
102 :
103 DATA   0         :REM   ZEILENANFANG
104 :
105 REM      INITIALISIERUNG
106 :
107 DATA 169,255     :REM   LDA #AUSGANG
108 DATA 141,  3,221:REM   STA DDR B
109 DATA 169,  0     :REM   LDA #INI
110 DATA 141,  1,221:REM   STA PORT B
111 DATA 169,125     :REM * LDA #CHROUTH
112 DATA 141, 39,  3:REM   STA CHRVEKH
113 DATA 141, 33,  3:REM   STA CHKVEKH
114 DATA 169, 55     :REM * LDA #CHROUTL
115 DATA 141, 38,  3:REM   STA CHRVEKL
116 DATA 169, 30     :REM * LDA #CHKOUTL
117 DATA 141, 32,  3:REM   STA CHKVEKL
118 DATA  96         :REM   RTS
119 :
120 REM      CHKOUT
121 :
122 DATA  32, 15,243:REM   JSR FILENR
123 DATA 240,  3     :REM   BEQ A
124 DATA  76,  1,247:REM   JMP F N OPEN
125 DATA  32, 31,243:REM A JSR SETPARA
126 DATA 165,186     :REM   LDA GERAETNR
127 DATA 201,  4     :REM   CMP #CEN
128 DATA 240,  5     :REM   BEQ B
129 DATA 165,186     :REM   LDA GERAETNR
130 DATA  76, 93,242:REM   JMP NOTCEN
131 DATA  76,117,242:REM B JMP CEN
132 :
133 REM      CHROUT
134 :
135 DATA  72         :REM   PHA
136 DATA 165,154     :REM   LDA GERAET
137 DATA 201,  4     :REM   CMP #CEN
138 DATA 240,  3     :REM   BEQ C
139 DATA  76,205,241:REM   JMP NOTCEN
140 DATA 134, 98     :REM C STX 98
141 DATA 132, 99     :REM   STY 99
142 DATA 169,255     :REM * LDA #CATL
143 DATA 133,100     :REM   STA 100
144 DATA 169,125     :REM * LDA #CATH
145 DATA 133,101     :REM   STA 101
146 DATA 160,227     :REM   LDY #LENCAT
147 DATA  36, 15     :REM   BIT LISTFLAG
148 DATA  16,  2     :REM   BPL X
149 DATA 160,255     :REM   LDY #LENCAT
150 DATA 162,  1     :REM X LDX #1
151 DATA 104         :REM   PLA
152 :
153 REM      SONDERZEICHEN
154 :
155 DATA 209,100     :REM D CMP (100),Y
156 DATA 240,  8     :REM   BEQ E
157 DATA 136         :REM   DEY
158 DATA 136         :REM   DEY
159 DATA 136         :REM   DEY
160 DATA 240, 30     :REM   BEQ G
161 DATA 136         :REM   DEY
162 DATA 208,244     :REM   BNE D
163 DATA 192,119     :REM E CPY #LISTCOD
164 DATA 144,  8     :REM   BCC F
165 DATA  36, 15     :REM   BIT LISTFLAG
166 DATA  48,  4     :REM   BMI F
167 DATA 169,  0     :REM   LDA #0
168 DATA  16, 55     :REM   BPL K
169 DATA 162,  3     :REM F LDX #3
170 DATA 136         :REM   DEY
171 DATA 177,100     :REM   LDA (100),Y
172 DATA  72         :REM   PHA
173 DATA 136         :REM   DEY
174 DATA 177,100     :REM   LDA (100),Y
175 DATA  72         :REM   PHA
176 DATA 136         :REM   DEY
177 DATA 177,100     :REM   LDA (100),Y
178 DATA  16, 40     :REM   BPL K
179 :
180 REM      ZEICHENSATZ ?
181 :
182 DATA  72         :REM G PHA
183 DATA 173, 24,208:REM   LDA VIDEO
184 DATA  41,  2     :REM   AND #2
185 DATA 208,  7     :REM   BNE H
186 :
187 REM      1.ZEICHENSATZ
188 :
189 DATA 104         :REM   PLA
190 DATA 201, 91     :REM   CMP #"Z"
191 DATA 144, 27     :REM   BCC K
192 DATA 176, 19     :REM   BCS Z
193 :
194 REM      2.ZEICHENSATZ
195 :
196 DATA 104         :REM H PLA
197 DATA 201, 64     :REM   CMP #ASCII
198 DATA 144, 10     :REM   BCC J
199 DATA 201,192     :REM   CMP #KLEIN
200 DATA 144,  4     :REM   BCC I
201 DATA  41,127     :REM   AND #7F
202 DATA 208,  2     :REM   BNE J
203 DATA   9, 32     :REM I ORA #32
204 DATA 201,123     :REM J CMP #"Z"
205 DATA 144,  6     :REM   BCC K
206 DATA 169, 32     :REM Z LDA #SPACE
207 DATA  72         :REM   PHA
208 DATA 232         :REM   INX
209 DATA 169, 96     :REM   LDA #JOKER
210 :
211 DATA  72         :REM K PHA
212 :
213 REM      RUECKTASTE
214 :
215 DATA 104         :REM L PLA
216 DATA 201,  6     :REM   CMP #RUECKT
217 DATA 208, 18     :REM   BNE O
218 DATA 206,  0,125:REM * DEC 32000
219 DATA  48, 53     :REM   BMI T
220 DATA 240,  9     :REM   BEQ N
221 DATA 169, 32     :REM   LDA #SPACE
222 DATA 232         :REM M INX
223 DATA  72         :REM   PHA
224 DATA 206,  0,125:REM * DEC 32000
225 DATA 208,249     :REM   BNE M
226 DATA 169, 13     :REM N LDA #CR
227 :
228 REM      DRUCKERCURSOR IN 32000
229 :
230 DATA 201, 13     :REM O CMP #CR
231 DATA 208,  5     :REM   BNE Q
232 DATA 160,  0     :REM P LDY #0
233 DATA 140,  0,125:REM * STY 32000
234 :
235 DATA 201, 27     :REM Q CMP #ESC
236 DATA 208,  3     :REM   BNE R
237 DATA 206,  0,125:REM * DEC 32000
238 :
239 DATA 201, 32     :REM R CMP #CONTROL
240 DATA 144, 23     :REM   BCC U
241 :
242 DATA 201, 94     :REM   CMP #AKZ.94
243 DATA 240,  4     :REM   BEQ S
244 DATA 201, 96     :REM   CMP #AKZ.96
245 DATA 208, 12     :REM   BNE T
246 DATA 168         :REM S TAY
247 DATA 169,  6     :REM   LDA #RUECKT
248 DATA  72         :REM   PHA
249 DATA 232         :REM   INX
250 DATA 169, 32     :REM   LDA #SPACE
251 DATA  72         :REM   PHA
252 DATA 232         :REM   INX
253 DATA 152         :REM   TYA
254 DATA 208,  3     :REM   BNE U
255 :
256 DATA 238,  0,125:REM T*INC 32000
257 :
258 REM      AUSGABE
```



```
259 :
260 DATA  72            :REM U PHA
261 DATA 173, 13,221:REM V LDA FLAG
262 DATA  41, 16     :REM   AND #16
263 DATA 240,249     :REM   BEQ V
264 DATA 104           :REM   PLA
265 DATA 141,  1,221:REM   STA PORT
266 DATA 202           :REM   DEX
267 DATA 208,175     :REM   BNE L
268 DATA 166, 98     :REM   LDX 98
269 DATA 164, 99     :REM   LDY 99
270 DATA  24           :REM   CLC
271 DATA  96           :REM   RTS
272 :
273 REM       OLYMPIADATEN
274 :
275 DATA  13, 10,  0, 13:REM CR MIT LF
276 DATA  13,  0,  0, 11:REM    OHNE LF
277 DATA  32,  0,  0,160:REM SHIFTSPACE
278 DATA  27, 54,  0,  1:REM 1 ZEILIG
279 DATA  27, 52,  0,  2:REM 1 1/2 ZG
280 DATA  27, 51,  0,  3:REM 2 ZEILIG
281 DATA  27, 80,  0, 21:REM 10 Z/ZEILE
282 DATA  27, 59,  0, 22:REM 12 Z/ZEILE
283 DATA  27, 77,  0, 23:REM 15 Z/ZEILE
284 DATA  27, 85,  0, 25:REM HZ VOR
285 DATA  27, 68,  0, 26:REM HZ RUECK
286 DATA 112,  6,105,255:REM ṗ
287 DATA  96, 94, 39, 94:REM ↑
288 DATA  62,  0,  0, 64:REM °
289 DATA  14, 47, 15, 62:REM >
290 DATA  14, 45, 15, 60:REM <
291 DATA 123,  0,  0,166:REM KL AE    "ä
292 DATA 124,  0,  0,220:REM KL OE    "ö
293 DATA 125,  0,  0,168:REM KL UE    "ü
294 DATA 126,  0,  0, 92:REM SZ       "ß
295 DATA  91,  0,  0,219:REM GR AE    "Ä
296 DATA  92,  0,  0,221:REM GR OE    "Ö
297 DATA  93,  0,  0,169:REM GR UE    "Ü
298 DATA  64,  0,  0,164:REM PARAGR   "§
299 DATA  14, 44, 15,186:REM TEILUNG "|
300 DATA  94,  0,  0,223:REM AKZ.94   "´
301 DATA  95,  6,  0,192:REM UNT+RT   "
302 DATA  14, 42, 15, 91:REM QUADRAT "²
303 DATA  14, 43, 15, 93:REM KUBIK    "³
304 DATA  95,  6,117,145:REM CURS U  "u
305 DATA  95,  6,100, 17:REM CURS D  "d
306 DATA  95,  6,108,157:REM CURS L  "l
307 DATA  95,  6,114, 29:REM CURS R  "r
308 DATA  95,  6, 99,147:REM CLR      "c
309 DATA  95,  6,104, 19:REM HOME     "h
310 DATA  95,  6,103,142:REM GROSS    "g
311 DATA  95,  6, 33,129:REM ORANGE   "!
312 DATA  95,  6, 34,149:REM BRAUN    ""
313 DATA  95,  6, 35,150:REM HELLROT "#
314 DATA  95,  6, 36,151:REM GRAU 1   "$
315 DATA  95,  6, 37,152:REM GRAU 2   "%
316 DATA  95,  6, 38,153:REM H.GRUEN "&
317 DATA  95,  6, 39,154:REM H.BLAU   "'
318 DATA  95,  6, 40,155:REM GRAU 3   "(
319 DATA  95,  6, 49,144:REM SCHWARZ "1
320 DATA  95,  6, 50,  5:REM WEISS    "2
321 DATA  95,  6, 51, 28:REM ROT      "3
322 DATA  95,  6, 52,159:REM CYAN     "4
323 DATA  95,  6, 53,156:REM PURPUR   "5
324 DATA  95,  6, 54, 30:REM GRUEN    "6
325 DATA  95,  6, 55, 31:REM BLAU     "7
326 DATA  95,  6, 56,158:REM GELB     "8
327 DATA  95,  6, 57, 18:REM RVS      "9
328 DATA  95,  6, 48,146:REM RVS OFF "0
329 DATA  95,  6, 78, 14:REM KLEIN    "N
330 DATA  95,  6, 72,  8:REM UMBLOCK "H
331 DATA  95,  6, 73,  9:REM UMFREI   "I
332 DATA  95,  6, 70,  6:REM RUECK    "F
333 DATA  95,  6, 75, 11:REM CR       "K
334 DATA  95,  6, 85, 21:REM 10 Z/Z   "U
335 DATA  95,  6, 86, 22:REM 12 Z/Z   "V
336 DATA  95,  6, 87, 23:REM 15 Z/Z   "W
337 DATA  95,  6, 89, 25:REM HZ VOR   "Y
338 DATA  95,  6, 90, 26:REM HZ RUECK"Z
```

Listing: Interface für Olympia Compact 2 (Schluß)



Ich bin ein Freund der guten Hires-Darstellungen. Mein Basic-Programm ermöglicht es, den Epson voll in seiner Fähigkeit des Plottens auszunutzen und das Bild zu verändern, das heißt:

1. Veränderung der Punktdichte
2. Veränderung der Darstellung

Für jede Veränderung müssen nur eine beziehungsweise mehrere Variablen geändert werden. Das Programm eignet sich natürlich auch für mathematische Funktionen oder ähnliches. Das Programm arbeitet problemlos mit dem Centronics-Interface von Data Becker.

(Mark Zimmermann/rg)

Listing: Hardcopy Epson FX-80

```
0 OPEN1,4,1:PRINT#1,CHR$(27);"L";CHR$(64
);CHR$(1);
2 DIM S(7,7):L=0:Q=8192
3 A=128:B=64:C=32:D=16:E=8:F=4:G=2:H=1
4 FORL=1TO25
6 FORR=QTOQ+320STEP8
8 FORN=0TO7
10 M=R+N
11 PRINTM:PRINT"□□"
12 X=PEEK(M)
28 IF X>=A THEN X=X-A:GOSUB 1000:S(N,0)=
1:GOTO 30
29 GOSUB 2000:S(N,0)=0
30 IF X>=B THEN X=X-B:GOSUB 1000:S(N,1)=
1:GOTO 50
40 GOSUB 2000:S(N,1)=0
50 IF X>=C THEN X=X-C:GOSUB 1000:S(N,2)=
1:GOTO 70
60 GOSUB 2000:S(N,2)=0
70 IF X>=D THEN X=X-D:GOSUB 1000:S(N,3)=
1:GOTO 90
80 GOSUB 2000:S(N,3)=0
90 IF X>=E THEN X=X-E:GOSUB 1000:S(N,4)=
1:GOTO110
100 GOSUB 2000:S(N,4)=0
110 IF X>=F THEN X=X-F:GOSUB 1000:S(N,5)
=1:GOTO130
120 GOSUB 2000:S(N,5)=0
130 IF X>=G THEN X=X-G:GOSUB 1000:S(N,6)
=1:GOTO150
140 GOSUB 2000:S(N,6)=0
150 IF X=HTHEN GOSUB 1000:S(N,7)=1:GOTO3
000
160 GOSUB 2000:S(N,7)=0
180 NEXTN:GOTO3000:REM*** WENN MEHR DATA
S DANN GOTO4! ***
1000 REM PRINT"*";
1010 RETURN
2000 REMPRINT".";
2010 RETURN
3000 A=0 :REM *** UMWANDLUNG DER WERTIKA
LEN DATAS IN HORIZONTALDATAS ***
4001 REM *** UMWANDLUNG DER HORIZONALEN
DATAS IN EPSON DATAS ***
4010 FORN=0TO7
4011 J=S(0,N)*128+(S(1,N)*64)+(S(2,N)*32
)+(S(3,N)*16+S(4,N)*8)
4015 K(N)=J+(S(5,N)*4)+(S(6,N)*2)+S(7,N)
4020 PRINT#1,CHR$(K(N));
4030 NEXTN:NEXTR:
4040 Q=Q+320:PRINT#1,CHR$(13);CHR$(27);"
L";CHR$(64);CHR$(1);:NEXTL
4050 REM *** BASIC HARDCOPY      ***
4060 REM *** BY MARK ZIMMERMANN ***
4070 REM *** POSTFACH 22 00 44 ***
4080 REM *** STADTWALDWENDE 20 ***
4090 REM *** D - 4300 ESSEN 01 ***
5000 REM *** FOR  C=64 & EPSON ***
READY.
```



Tips & Tricks

Listschutz

Möchte man ein Programm mit einem einfachen Listschutz versehen, so verfährt man folgendermaßen:

1. Man ergänzt die Zeile, ab der der Listschutz wirksam werden soll, mit »:REM" "«.
2. Man fährt mit dem Cursor auf das zweite Anführungszeichen und drückt fünfmal die Taste INST.
3. Nun wird ebenfalls fünfmal die Taste DEL gedrückt, so daß zwischen den Anführungszeichen fünf reverse T stehen.
4. Zuletzt bewegt man den Cursor hinter das zweite Anführungszeichen und drückt die Tastenkombination SHIFT und L. Anschließend RETURN nicht vergessen.

Wenn nun versucht wird, das Programm zu listen, gelangt der Computer nur bis zu der Zeile, in der der Listschutz steht und bricht dann den Vorgang mit »Syntax Error« ab.

(Thomas Lopatic)

Für alle diejenigen C 64 - Besitzer, denen die Bewegung des Cursors bisher zu langsam war, gibt es einen speziellen POKE.

Mit POKE 56325,5 wird der Cursor rasend schnell und flitzt bei Betätigung der Cursortasten nur noch so über den Bildschirm. Wer's lieber gemütlicher mag, der sollte es statt dessen einmal mit POKE 56325,255 probieren.

(Oliver Bausch)

Im Leserforum des 64'er Magazins, Ausgabe 8/84, fragte Frank Schager nach einer Möglichkeit, mit Simons Basic ein Textfenster in der hochauflösenden Grafik zu erzeugen.

Mit dem folgenden kleinen Programm wird die normale Tastaturabfrage mit dem Simons Basic-Befehl »TEXT« verbunden:

```
10 X = 2 : Y = 2 : REM Text-Anfang
20 HIRES 15, 11 : REM Grafik ein
30 GET A$
40 IF A$ < > " "THEN GOSUB 100
50 GOTO 30
60 REM
100 X = X + 8 : REM X-Koordinate erhöhen
110 IF X > 38 * 8 THEN X = 2 : Y = Y + 8 : REM Zeilenen-
de ? Dann neue Zeile
120 TEXT X,Y,A$,1,1,8 : REM Zeichen drucken
130 AA$ = AA$ + A$ : REM Wort erzeugen
140 IF AA$ = "GEHE" THEN 1000 : REM Zum Beispiel
150 RETURN
```

Natürlich kann man die Anfangskoordinaten so verändern, daß sie zum Spiel passen. Veränderbar ist auch der Faktor 38 in Zeile 110, je nachdem, welche Zeilenlänge gewünscht wird. Ebenso besteht die Möglichkeit, zwischen den Zeilen 130 und 150 weitere IF-Abfragen einzubauen. Empfehlenswert ist die besondere Abfrage der Tasten SPACE und RETURN.

(Jörg Prante)

Bei Ihrem Software-Test des Textomat von Data-Becker (Ausgabe 9/84) wurde als gravierender Nachteil angeführt, daß bei einmal gestartetem Ausdruck keine Unterbrechung mehr möglich ist. Ich arbeite in Zusammenhang mit Textomat mit einem Epson-Drucker RX 80 F/T. Um den begonnenen Ausdruck zu unterbrechen, kann man einfach den ON-LINE-Schalter betätigen und den Drucker anschließend ausschalten. Sekunden später meldet sich der Textomat am Bildschirm mit dem zu druckenden Text zurück.

(H. Rendelmann)

So manch einen C 64-Besitzer wird es schon geärgert haben, daß sein Computer keinen MERGE-Befehl besitzt. Mit wenig Aufwand ist es aber dennoch möglich, Basic-Programme aneinanderzuhängen:

1. Im Direktmodus »PRINT PEEK(43); PEEK(44)« eingeben und sich die Ergebnisse merken.
2. Das erste Programm normal laden.
3. Erscheint jetzt nach »PRINT PEEK(45)« eine Null oder eine Eins, dann geben Sie »POKE 43, 256 + PEEK(45) - 2 : POKE 44, PEEK(46) - 1 : NEW« ein. Im anderen Fall wird »POKE 43, PEEK(45) - 2 : POKE 44, PEEK(46) : NEW« eingegeben.
4. Nun wird das anzuhängende Programm geladen (Achtung! Das anzuhängende Programm muß die höheren Zeilennummern haben).
5. Jetzt POKEn Sie in die Speicherstellen 43 und 44 die zu Anfang gemerkten Werte.

Beide Programme sind nun verbunden und können ganz normal gehandhabt werden. Wichtig bei der ganzen Prozedur ist, daß keine Variablen definiert werden, da das MERGEn sonst nicht richtig funktioniert.

(Thomas Lopatic)

POKEs für den C 64

Mit POKE 775, 1 ist ein (fast) perfekter Listschutz aktiviert. Auch ein SAVE-Schutz ist mit wenig Aufwand möglich: POKE 801,0 : POKE 802,0 : POKE 818,165. Nach diesen drei POKE-Befehlen kann das Programm weder auf Kassette noch auf Diskette kopiert werden. Schließlich gibt es noch eine Möglichkeit, die RUN/STOP-Taste abzuschalten, und zwar mit POKE 808,225. Wiedereinschalten ist mit POKE 808,237 möglich.

(Thomas Lopatic)

Viele schätzen es, viele wünschen es sich: Einen berechneten GOTO-Befehl auf einen variablen Ausdruck anstelle einer Zeilennummer. Hier ist eine schnelle und sichere Methode, die nur 17 Bytes Speicherplatz benötigt.

Schreiben Sie als erste Programmzeile »1 REM "*********"« (mindestens neun Sternchen). Anschließend geben Sie im Direktmodus ein:

»PRINT PEEK(43) + PEEK(44) * 256 + 6« (RETURN). Die daraufhin angezeigte Adresse notieren Sie sich bitte. Ohne Erweiterung müßten Sie den Wert 4103 erhalten haben, mit 3 KByte Erweiterung 1031 und ab 8 KByte Erweiterung 4615.

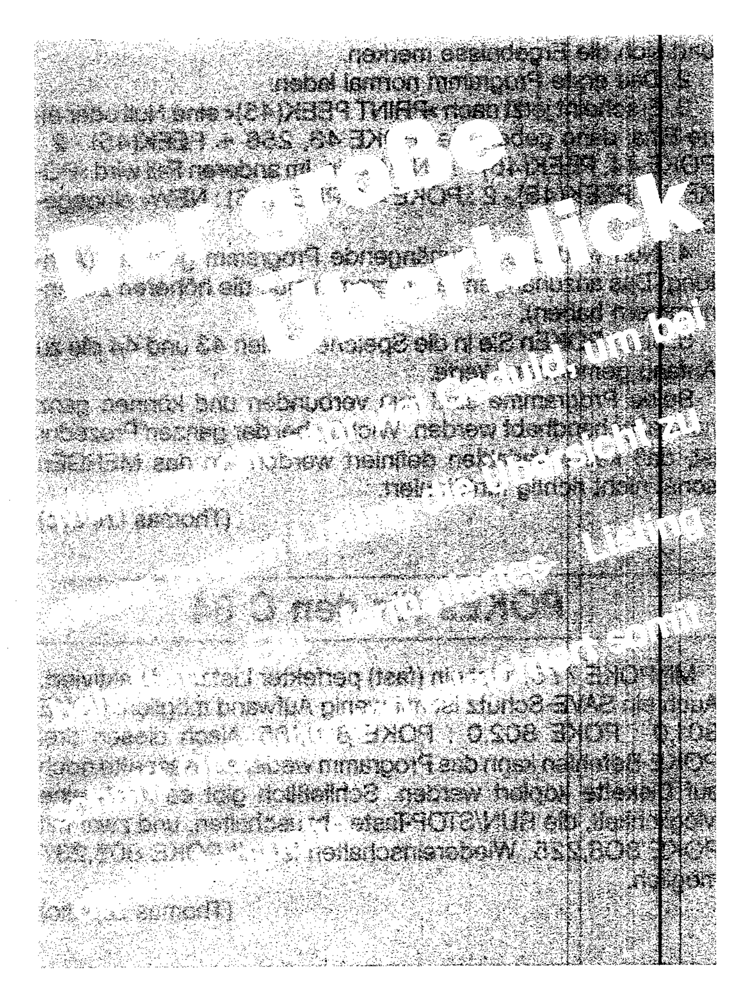
Jetzt POKEn Sie ab der notierten Adresse bitte folgende Werte ein: 32, 138, 205, 32, 247, 215, 76, 163, 200.

Wenn Sie nun die erste Zeile (mit dem REM) aufLISTen, sehen Sie einige Grafikzeichen. Diese stellen ein kurzes Maschinenspracheprogramm dar, das einen mathematischen Ausdruck in einen ganzzahligen Wert umrechnet. Diese Zeile muß immer die erste Programmzeile sein und darf auch nicht mehr geändert werden. Das übrige Programm kann natürlich wie gewohnt editiert werden.

Sie haben jetzt im Programm einen simulierten GOTO X - Befehl zur Verfügung, der mit SYS (Adresse) X aufgerufen wird. Für Adresse müssen Sie die anfangs notierte Adresse einsetzen (Klammern nicht vergessen). Für X kann ein beliebiger arithmetischer Ausdruck stehen wie zum Beispiel 5, A, A+2, C+D/SQR(9) oder PEEK(5).

Der neue Befehl hat im übrigen die gleichen Auswirkungen wie der normale GOTO-Befehl. Ist eine Zeilennummer nicht vorhanden, gibt es daher ebenfalls einen »UNDEF'D STATEMENT ERROR«.

(Thomas Maul)



Das Programm FLIST belegt 134 Bytes im Kassettenpuffer des C 64. Es wird mit LOAD"FLIST.DATA", 8 geladen und mit RUN gestartet. Dabei wird das , in DATA-Zeilen stehende Programm im Speicher ab 828 abgelegt und mit SYS gestartet. Die Zeile 10, die diese Dinge erledigt, dient später als Demonstrationszeile. Nun zur Funktion:

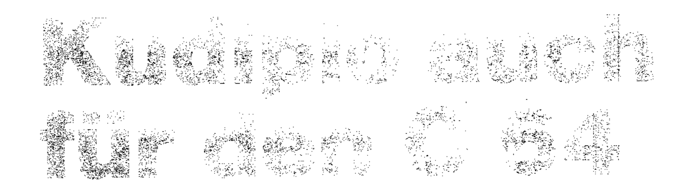
Das Programm erweitert das Basic um den Befehl FLIST. Dieser Befehl erzeugt ein formatiertes Listing auf dem Bildschirm, indem vor der Ausgabe eines Zeichens geprüft wird, ob dieses Zeichen ein »:« ist. Trifft dies nicht zu, so wird das 'LISTen' fortgesetzt. Andernfalls erzeugt das Programm ein »CR« (Carriadge Return) und fügt eine, von der Länge der Zeilennummer abhängige Anzahl von Leerzeichen ein. Dadurch stehen alle in einer Zeile vorhandenen Doppelpunkte untereinander und unter dem Leerzeichen nach der Zeilennummer. Jeder Befehl, vor dem ein Doppelpunkt steht, erscheint so in einer neuen Zeile. Die Angabe der Zeilennummer(n) erfolgt bei FLIST wie beim normalen LIST-Befehl (zum Beispiel FLIST-100). Zum Editieren einer Zeile steht dem Benutzer weiterhin der normale LIST-Befehl zur Verfügung. Versuchen Sie mal FLIST10! Benutzer eines Druckers könnten ja mal OPEN1,4:FLIST:close1 ausprobieren.

(Michael Weidlich/rg)

```
10 FORI=828TO962::    READJ:    POKEI,J:
:NEXTI::SYS828::END
100 DATA169,71,141,8
110 DATA3,169,3,141
120 DATA9,3,96,32
130 DATA115,0,201,155
140 DATA240,10,201,70
150 DATA240,25,32,121
160 DATA0,76,231,167
170 DATA169,26,141,6
180 DATA3,169,167,141
190 DATA7,3,32,115
200 DATA0,32,156,166
210 DATA76,174,167,32
220 DATA115,0,201,155
230 DATA240,3,76,8
240 DATA175,169,130,141
250 DATA6,3,169,3
260 DATA141,7,3,76
270 DATA98,3,8,36
280 DATA15,48,4,201
290 DATA58,240,4,40
300 DATA76,26,167,72
310 DATA138,72,152,72
320 DATA160,2,177,95
330 DATA133,99,200,177
340 DATA95,133,98,162
350 DATA144,56,32,73
360 DATA188,32,223,189
370 DATA32,135,180,32
380 DATA166,182,170,169
390 DATA13,32,12,225
400 DATA32,59,171,202
410 DATA208,250,104,168
420 DATA104,170,104,40
430 DATA76,26,167
READY.
```

```
PRUEFSUMMENLISTE
BLOCKGROESSE 20
-----------------------------------------------
ZEILE  ANZAHL   SUMME     KEIN POKE?
-----------------------------------------------
  140     20     1837
  190     40     3708
  240     60     5924
  290     80     7087
  340    100     9393
  390    120    11679
-----------------------------------------------
GESAMT   135    13559
```





In der Ausgabe 8 des 64'er Magazins war das Programm »Kudiplo« für den VC 20 abgedruckt, das mit dem 1520-Printer-Plotter eine komplette Kurvendiskussion ausgibt. Hier sind die erforderlichen Änderungen, um dieses nützliche Programm auch auf dem C 64 laufen zu lassen.

Nach der Veröffentlichung meines Programms Kudiplo für den VC 20, erreichten mich viele Leserbriefe mit der Bitte um nähere Auskunft dazu, wie das Programm für den C 64 abzuändern ist. Probleme gab es dabei mit der Routine in den Zeilen 185, 190 und 230.

Die genannten Zeilen bewirken in der veröffentlichten Version für den VC 20 ein »Verbiegen« des Vektors für die Fehlerbehandlungsroutine. Der in den Speicherstellen $0300 und $0301 stehende Zeiger wird so verändert, daß er nicht mehr zu der im Basic-ROM stehenden Routine zur Ausgabe von Fehlermeldungen zeigt. Statt dessen zeigt er nun auf einen Sprungbefehl, der mit Hilfe der Zeile 185 in den Kassettenpuffer geschrieben wurde. Dieser Sprungbefehl führt zurück ins Basic-Programm, dessen nächste Zeile gesucht und so abgearbeitet wird, als sei kein Fehler aufgetreten.

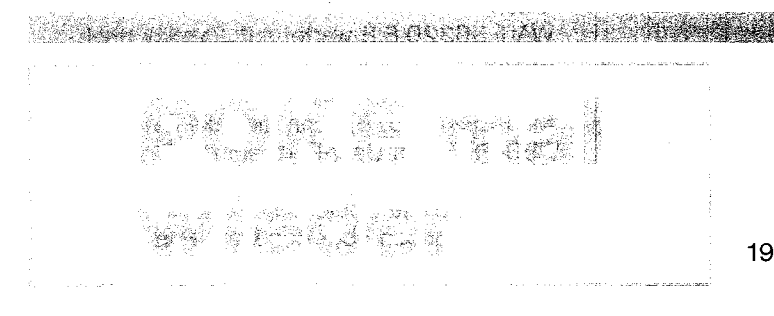
Die Ähnlichkeit des VC 20 mit seinem großen Bruder ist oft zitiert. Auch bei ihm läßt sich eine solche Fehlerblockade einrichten. Allerdings ist beim großen Bruder zu diesem Zweck ein kleines Maschinenprogramm in den Kassettenpuffer zu schreiben, in welchem abgefragt wird, ob ein Fehler vorgekommen ist und das dann abhängig vom Ergebnis entweder zum nächsten Statement oder zur nächsten Zeile verzweigt.

Für den C 64 müssen darum die Zeilen 185 und 230 wie folgt geändert werden:

```
185 DATA 138, 48, 3, 76, 59, 169, 76, 116, 164 : FOR I=823
TO 840 : READ A : POKE I, A : NEXT
230 NEXT : POKE 768, 139 : POKE 769, 227
```

Mit diesen Änderungen läuft Kudiplo dann endlich auch auf dem »großen Bruder«.

(Jürgen Curdt/ev)



Viele C 64-Benutzer haben sich sicher schon mit dem Basic des C 64 herumgeärgert: Egal, was man machen will, fast alles läuft über PEEK und POKE. Doch gerade diese POKEs helfen manchmal erheblich, wenn es um Probleme geht, die mit einfachen Basic-Befehlen nicht zu lösen sind.

Hier nun eine Liste von wichtigen PEEKs, POKEs und SYS-Befehlen.

1	Inhalt 55 = normal Inhalt 54 = Basic ausgeschaltet (auf RAM umgestellt) Inhalt 53 = Basic und Kernal auf RAM umgestellt. (Es empfiehlt sich dabei, das Basic und das Kernal vorher ins RAM zu POKEn, damit der Computer bei der Umschaltung nicht aussteigt.)
17	Mit diesem PEEK läßt sich abfragen, wie die letzte Variable zugewiesen wurde. Ist PEEK(17) = 00, dann war die letzte Variablenzuweisung ein INPUT, oder es hat noch keine Zuweisung stattgefunden. Ist PEEK(17) = 64, dann wurde die letzte Variable durch GET geholt. Bei PEEK(17) = 152 erfolgte die letzte Variablenübergabe durch einen READ-Befehl.
19	Durch POKE 19,64 wird beim nächsten INPUT-Befehl kein Fragezeichen mehr ausgegeben. Allerdings kann man nachher durch Drücken der RETURN-Taste nicht mehr in die nächste Zeile gelangen. Es empfiehlt sich daher, nach dem INPUT-Befehl diesen Befehl wieder mit POKE 19,0 rückgängig zu machen.
43/44	Der Anfang des zur Zeit im Speicher befindlichen Basic-Programms errechnet sich durch PEEK(43)+PEEK(44)*256.
45/46	Das Ende des Basic-Programms erhält man durch ?PEEK(45)+PEEK(46)*256.
61/62	Zeiger auf Basic-Statement für CONT: Durch PEEK(61)+PEEK(62)*256 erhält man die Speicherstelle, die nach dem zuletzt ausgeführten Basic-Befehl liegt, das heißt die Speicherstelle, von der sich der Basic-Interpreter bei CONT den nächsten Befehl holt. Tip: Bei CONT kommt öfter CAN'T CONTINUE ERROR vor, wenn man nach dem Stoppen ein CLR eingegeben oder in irgendeiner Programmzeile etwas geändert hat. Liest man die Werte mit PEEK(61) und PEEK(62) nach der Unterbrechung aus, dann macht ein CLR oder ähnliches nichts aus, wenn man vor CONT die zuvor ausgelesenen Werte wieder in die Speicherstellen POKEt.
63/64	Nummer der aktuellen DATA-Zeile: Mit ?PEEK(63)+PEEK(64)*256 erhält man die Nummer der DATA-Zeile, aus der gerade das letzte Datum geholt wurde. (Gut zum Finden von Fehlern in DATA-Zeilen geeignet.)
69/70	Name der zuletzt zugewiesenen Variable: Bei normalen Fließkommavariablen liest man

den Wert mit PRINT CHR$(PEEK(69)) + CHR$(PEEK(70)) aus.
Bei Integervariablen (zum Beispiel XY%) erhält man den Namen durch ?CHR$(PEEK(69)-128)+CHR$(PEEK(70)-128).
Strings (zum Beispiel VX$) erhält man durch ?CHR$(PEEK(69))+CHR$(PEEK)70)-128).

120 Nach Ausführung dieses POKEs nimmt der C 64 keinerlei Befehle mehr an:
POKE 120,2

147 Wenn man die LOAD-Routine im Betriebssystem anspringt, holt es sich aus der Speicherstelle 147 die Information, ob LOAD oder VERIFY durchgeführt wird.
Inhalt 0 = LOAD
Inhalt 4 = VERIFY

157 Ausgabe-Kontrolle:
Inhalt 000 = Programm-Modus
Inhalt 128 = Direktmodus
Damit bei LOAD-Befehlen vom Programm aus die Mitteilungen SEARCHING, LOADING oder VERIFYING auf dem Bildschirm erscheinen, setzt man vor dem LOAD-VERIFY- oder SAVE-Befehl ein POKE157,128.

197 Derzeitiger Tastendruck:
PEEK(197)

200 Zeiger auf Zeilenende. PEEK(200) gibt an, wieviel Zeichen die zuletzt eingegebene Zeile hatte.

204 Nach POKE 204,0 bleibt der Cursor an, auch bei GET-Befehlen. Mit POKE 207,0:POKE 204,1 kommt man dann wieder auf den Normalzustand zurück.

641-644 Start- und Endadresse des Basic-RAMs: Durch Ändern dieser Werte kann man die Größe des Basic-RAMs verändern, zum Beispiel:
POKE 643,0 : POKE 644,128 : SYS 64764 setzt das Ende des Basic-RAMs um 8 KByte nach unten. Anderes Beispiel: POKE 641,0 : POKE 642,16 : SYS 64764 setzt das Basic-RAM um 2 KByte nach oben.

646 POKE 646,Farbwert setzt die Cursorfarbe.

653 PEEK für Shift-, Commodore- und für CTRL-Taste:
Bit 0 = Shift-Taste, Bit 1 = Commodore-Taste und Bit 2 = Control-Taste.

788/789 IRQ, Hardware-Interrupt: Das Betriebssystem springt ständig in diese Routine, durch Ändern des Inhalts kann man eigene, »interruptgesteuerte« Maschinenroutinen ständig laufen lassen.

792/793 Restore-Vektor: PEEK(792) + PEEK(793)*256 ergibt die Speicherstelle, an die bei Restore-Tastendruck gesprungen wird.
Beispiel: Bei POKE 792,226 : POKE 793,252 wird bei Drücken der Restore-Taste ein Reset ausgelöst.

828-1019 Kassettenpuffer: Nach Laden oder Verify stehen im Kassettenpuffer folgende Informationen:
828: 1 = normales File, 3 = wurde mit SAVE"Name",1,1 abgespeichert. Solche Programme werden bei LOAD automatisch ab der Adresse geladen, von der sie abgespeichert wurden.
829/830: Hier ist die Startadresse des Programms abgelegt (829 ist das Low-Byte, 830 das High-Byte).
831/832: Endadresse des Programms.
833-1019: 186 Zeichen langer Programmname (auf dem Bildschirm werden nur 16 angezeigt, aber es lassen sich bis 186 Stellen lange Programmnamen abspeichern).
Der Kassettenpuffer ist auch gut zum Ablegen eigener Maschinenprogramme geeignet, sofern mit der Floppy gearbeitet wird.

42291 Koppeladressen angleichen: Falls Programme mit NEW gelöscht wurden, kann man mit diesem SYS-Befehl die Bytes 2049 und 2050 wieder in Ordnung bringen, wenn vorher etwas anderes als 0 in diese Speicherzellen gePOKEt wird.

56320 Joystick Port 2:
WAIT 56320,16,16 wartet auf Feuerknopf
WAIT 56320,4,4 wartet auf Linksbewegung des Joysticks
WAIT 56320,1,1 wartet auf Joystick nach oben
WAIT 56320,2,2 wartet auf Joystick nach unten
WAIT 56320,8,8 wartet auf Joystick nach rechts

56321 Wie 56320, aber Joystick in Port 1.

56576 Mit PEEK(56576) kann man die Pins PB0-PB7 vom User-Port (auf der Unterseite des Ports, siehe Handbuch) auslesen. Mit POKE in diese Speicherstelle kann man auch Ausgaben über den User-Port laufen lassen.

56578 Datenrichtungsregister für User-Port: Jedes der Bits gibt die Datenrichtung für die Pins PB0-PB7 des User-Ports an. Ist das entsprechende Bit gesetzt, so fungiert der dem Bit zugeordnete Pin als Ausgang, bei nicht gesetztem Bit als Eingang.

65409 SYS 65409 setzt den Video-Chip des C 64 auf den Ursprungszustand zurück.

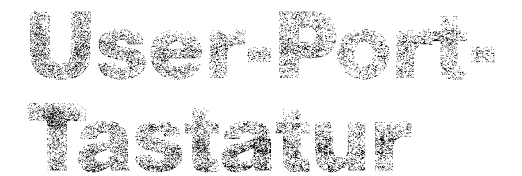
65493 LOAD-Routine des Betriebssystems. Mit folgender kleiner Routine kann man Unterprogramme nachladen, ohne irgendwelche Basic-Pointer (wie zum Beispiel die Zeiger auf die Endadresse, 45 und 46) zu verändern:
POKE 186,1 : POKE 780,0 :
POKE 781,0 : POKE 782,96 :
POKE 183,0 : SYS 65493
Erklärung: 186,1 = Geräteadresse für Recorder
781 und 782 gibt die Startadresse an, ab der das Programm geladen werden soll.
183,0 = kein Programmname.
SYS 65493 = LOAD-Routine.

65511 Durch SYS 65511 lassen sich alle Files schließen. So erspart man sich das lästige Eintippen von CLOSE1:CLOSE2:CLOSE3... Dabei sollte aber beachtet werden, daß so nur der Kanal geschlossen wird, aber keine Dateien auf einer Disk.

Auf PEEKs und POKEs für Grafik und Sprites wurde hier verzichtet, da die Grafik und die Sprites im Grafikkurs von H. Ponnath schon sehr ausführlich beschrieben sind.

(M. Kohlen/gk)





Mit Hilfe dieses Programmes ist es möglich, eine beliebige Tastatur mit bis zu 4 x 5 Matrix an den User-Port anzuschließen und die Funktionstasten mit Strings bis 9 Byte Länge zu belegen.

Bei der Auswahl einer geeigneten Tastatur gibt es eigentlich nur eine Bedingung: Sie darf höchstens 4 x 5 Anschlüsse haben und die Tasten müssen auf den Kreuzungspunkten der 4 x 5 Anschlüsse liegen. Ansonsten gibt es keine Beschränkungen, da jede Taste frei programmierbar ist. Um die Tastatur zu programmieren, muß man über ihren Aufbau nichts wissen. Man schließt sie einfach am User-Port an den Leitungen Pb0 bis Pb7 und Pa2 an. Die Anschlüsse Pb0 bis Pb3 sind die Zeilen der Tastatur. Es sind am User-Port die Anschlüsse Pb0=c, Pb1=d, Pb2=e, Pb3=f. Für die Spalten gelten die Anschlüsse Pb4=h, Pb5=j, Pb6=k, Pb7=l, Pa2=m.

Die Programmierung geschieht mit Hilfe des Basic-Ladeprogramms. Man wird gebeten, die zu programmierende Taste zu drücken. Die momentane Belegung wird ausgegeben, und man wird gefragt, welchen ASCII-Wert man der Tastatur zuweisen möchte. Danach gibt man den entsprechenden ASCII-Wert ein. Danach wird man erneut gebeten, eine Taste zu drücken. Es ist also nicht notwendig, die Matrix zu kennen. Fertig programmierte Tastaturen können dann als Maschinenprogramm auf Diskette abgespeichert werden, welches nur zwei Blöcke belegt. Es kann dann von Diskette mit LOAD »name«,8,1 geladen und mit SYS12*4096 gestartet werden. Im Zusammenhang mit der Funktionstastenbelegung können auf die Tastatur auch ganze Strings gelegt werden, indem man nur den entsprechenden ASCII-Code der Funktionstasten (133-139) auf die Zusatztastatur programmiert. Auf diese Weise ist es möglich, komplette DATA-Listings mit einer kleinen Tastatur mit Zehnerblock bequem und schnell abzutippen, da man auch die Returntaste und alle Sonderzeichen auf die Tastatur umlegen kann. Auch eine komplette Hex-Tastatur mit den Zeichen 0-9 und a-f und vier Sonderzeichen, zum Beispiel Return, Delete, Komma, Punkt (Grundausstattung), kann programmiert werden. Dadurch wird das Abtippen von Hexdumps zum Kinderspiel. Bewährt hat sich eine alte Rechenmaschinentastatur, mit der ich auch schon Spiele gespielt habe.

Die Belegung der Funktionstasten geschieht ebenfalls durch das Ladeprogramm. Zunächst wird gezeigt, wie die Tasten belegt sind und gefragt, ob man die Belegung ändern will. Man gibt dann nur den entsprechenden String ein und drückt Return. Zum Schluß wird man noch gefragt, ob man mit seiner Wahl zufrieden ist.

(Martin Kloss/aa)

Pb4 Pb5 Pb6 Pb7 Pa2

Pb0

Pb1

Pb2

Pb3

Jeder Kreuzungspunkt stellt eine Taste dar.

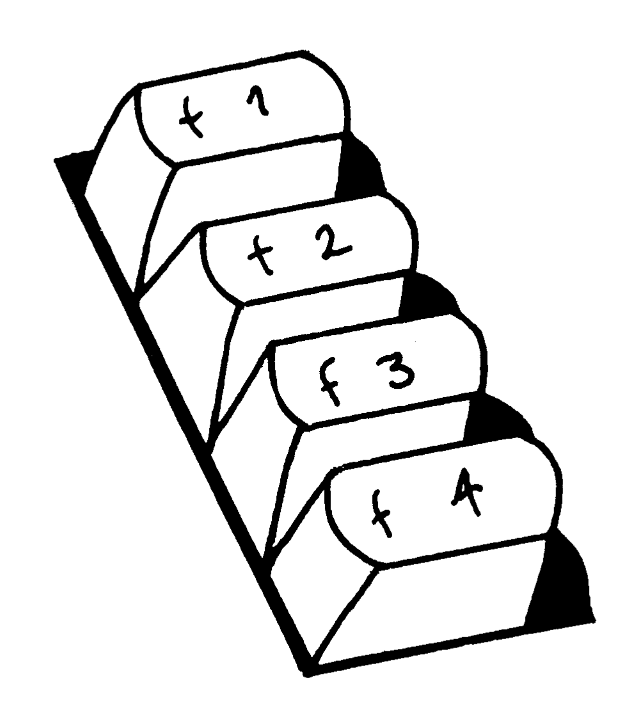
```
100 REM" -------------------------------------
110 REM"|USERPORT UND FUNKTIONSTASTEN  |
120 REM"|BELEGUNG FUER DEN C 64    VON |
130 REM"|MARTIN KLOSS    DUESSELDORF   |
140 REM"|------------------------------|
150 REM"|NN,MM   :LAUFVARIABLE         |
160 REM"|Q1,Q2   :READ,AENDERUNGSVAR.  |
170 REM"|Q1$,Q2$:LOESCH,ANTWORTVAR.    |
180 REM"|Q2$( )  :BEFEHLSFELD          |
190 REM"|------------------------------|
200 REM"|ANSCHLUSS EINER TASTATUR MIT  |
210 REM"|4*5 MATRIX AM USERPORT        |
220 REM"|4 ZEILEN AUF PB0-PB3          |
230 REM"|5 SPALTEN AUF PB4-PB7 U. PA2  |
240 REM"|20 PROGRAMMIERBARE TASTEN     |
250 REM"|------------------------------|
260 REM"|BELEGUNG DER FUNKTIONSTASTEN  |
270 REM"|MIT STRINGS BIS 9 BYTE LAENGE |
280 REM"|FREI PROGRAMMIERBAR           |
290 REM"|------------------------------
300 PRINT"          "SPC(10)"--------------------
"
310 PRINTSPC(10)"  USERPORTTASTATUR "
320 PRINTSPC(10)"                   "
330 PRINTSPC(10)"  BY MARTIN KLOSS  "
340 PRINTSPC(10)" -------------------"
350 B=0:FORNN=49152TO49527:READA:B=B+A:P
OKENN,A:NEXTNN:SYS49152
360 IFB<>43253THENPRINTSPC(10)"     DATA
ERROR ":STOP
370 FORNN=0TO7:READQ2$(NN):NEXTNN:PRINT"
 ":Q1$=" *                     "
380 PRINTSPC(5)"-------------------------
"
390 PRINTSPC(5)"  FUNKTIONSTASTENBELEGUN
G "
400 PRINTSPC(5)" ------------------------
```

Listing für die User-Port-Tastatur

```
"
410 PRINTSPC(5)"JETZIGE BELEGUNG"
420 FORNN=0TO7:READQ1:PRINTSPC(5)"F"Q1"
  "Q2$(NN):NEXTNN
430 GOSUB670
440 IFQ3$<>"J"THENPOKE58,255:GOTO530
450 PRINT"":FORNN=0TO7:READQ1
460 PRINTSPC(3)"FUNKTIONSTASTE "Q1" "Q2$
(NN):PRINT""TAB(20);
470 INPUTQ2$(NN):IFLEN(Q2$(NN))>9THENPRI
NTTAB(20)Q1$:PRINT""TAB(20);:GOTO470
480 NEXT NN
490 PRINTSPC(3)"EINGABEN KORREKT (J)";:
INPUTQ3$:IFQ3$<>"J"THEN RUN
500 Q2=49420:FORNN=0TO7:FORMM=1TOLEN(Q2$
(NN))
510 POKEQ2+MM,ASC(MID$(Q2$(NN),MM,1)):NE
XTMM
520 FORMM=LEN(Q2$(NN))+1TO9:POKEQ2+MM,0:
NEXTMM:Q2=Q2+10:NEXTNN:POKE58,255
530 PRINT""SPC(5)"
"
540 PRINTSPC(5)"  TASTATURBLOCKBELEGUNG
 "
550 PRINTSPC(5)"
"
560 GOSUB670:IFQ3$<>"J"THEN680
570 PRINTSPC(5)"BITTE TASTE DIE GEAEND
ERT                WERDEN SOLL DRUECKEN"
580 Q2=PEEK(49402):IFQ2=239THEN580
590 Q3=PEEK(49501+Q2):PRINTSPC(5)"TASTE
 BELEGT MIT CHR$("Q3") ";:
600 IFQ3<128ANDQ3>32THENPRINT" = "CHR$(Q
3);
610 IFPEEK(49404)=Q3THENGETQ3$:GOTO610
620 PRINT:FORNN=0TO20:GETQ3$:NEXT
630 PRINTSPC(5)"WELCHEN ASC 2 WERT SOLL
 DIE":INPUT"      TASTE ERHALTEN";Q3$
640 IFVAL(Q3$)>255THENPRINTSPC(5)"KEIN A
SC 2 ZEICHEN ":GOTO630
650 POKE(49501+Q2),VAL(Q3$)
660 PRINTSPC(3)"WEITERE";:GOTO560
670 PRINTSPC(5)"AENDERUNG ERWUENSCHT (J
)";:INPUTQ3$:RETURN
680 PRINT""SPC(5)"SOLL DIE GEAENDERTE
 VERSION ALS             MASCHIENENPROGRAMM"
690 PRINTSPC(5)"AUF DISK ABGESPEICHERT W
ERDEN";:INPUTQ3$:IFQ3$<>"J"THENEND
700 PRINTSPC(5)"DATEINAME ";:INPUTQ3$
710 OPEN8,8,1,"@:"+Q3$+",P,W"
720 PRINT#8,CHR$(0);:PRINT#8,CHR$(192);
730 FORNN=49152TO49527:PRINT#8,CHR$(PEEK
(NN));:NEXTNN:CLOSE8:END
740 DATA120,169,15,141,3,221,173,2,221,9
,4,141,2,221,169,219,141,0,3,169
750 DATA192,141,1,3,169,36,141,20,3,169,
192,141,21,3,88,96,36,157,16,48,169
760 DATA133,141,248,192,165,215,141,247,
192,173,247,192,205,248,192,240,12
770 DATA238,248,192,173,248,192,201,141,
240,20,208,236,56,173,247,192,233
780 DATA133,10,170,189,254,192,168,189,2
53,192,32,30,171,169,254,141,250
790 DATA192,162,4,56,169,0,141,251,192,1
73,250,192,141,1,221,173,1,221,205
800 DATA1,221,208,248,141,249,192,24,160
,4,14,249,192,144,35,136,208,248
810 DATA169,4,141,251,192,173,0,221,41,4
,240,20,169,0,141,251,192,56,46,250
820 DATA192,202,208,198,169,0,141,252,19
2,76,49,234,142,249,192,140,250,192
830 DATA173,251,192,109,250,192,141,250,
192,174,249,192,240,15,173,250,192
840 DATA105,5,141,250,192,206,249,192,20
8,238,240,236,174,250,192,189,93
850 DATA193,205,252,192,240,207,141,252,
192,170,32,33,235,24,144,197,176
860 DATA195,120,169,15,141,3,221,173,2,2
21,9,4,141,2,221,169,36,141,20,3
870 DATA169,192,141,21,3,88,76,139,227,6
9,141,112,239,0,0,13,193,23,193,33
880 DATA193,43,193,53,193,63,193,73,193,
83,193,28,85,78,0,0,0,0,0,0,76
890 DATA73,83,84,0,0,0,0,0,0,76,79,65,68
,0,0,0,0,0,0,68,65,84,65,0,0,0,0
900 DATA0,0,86,69,82,73,70,89,0,0,0,0,83
,65,86,69,0,0,0,0,0,0,67,76,79,83
910 DATA69,0,0,0,0,0,79,80,69,78,0,0,0,0
,0,0,85,86,87,88,89,90,20,44,48,69
920 DATA67,49,50,51,70,13,52,53,54,46,66
,55,56,57,65,68,0,RUN,LIST,LOAD,DATA
930 DATAVERIFY,SAVE,CLOSE,OPEN,1,3,5,7,2
,4,6,8,1,3,5,7,2,4,6,8

READY.
```

Listings für die User-Port-Tastatur



Diskette intern

Der genaue Inhalt einer Diskette ist den meisten Floppy-Besitzern wohl bisher unbekannt geblieben. Das Programm Disk-Dump schafft den Einblick.

Dieses Programm ermöglicht den Inhalt einzelner Disketten-Blocks auf dem Bildschirm anzusehen, sie auszudrucken und zu ändern!

Nach dem Start des Programms wird Track 18 — Sektor 0 in den Speicher geladen und — in zwei Stufen — auf den Bildschirm ausgegeben. Zwei Stufen sind notwendig, um den Inhalt gut auf dem Bildschirm darzustellen. In der ersten Stufe werden die ersten 128 Byte angezeigt, je 8 Byte in einer Zeile. Am Anfang der Zeile steht noch die Hexnummer des ersten Bytes der jeweiligen Zeile, hinter dem Inhalt werden die 8 Byte noch als Character-String ausgegeben. Hierbei werden Steuerzeichen als Blank ausgegeben. Dieser Bildschirminhalt bleibt nun bestehen, bis eine Taste gedrückt wird und somit die zweiten 128 Byte in gleicher Weise angezeigt werden.

Im Anschluß daran erscheint, zusammen mit den letzten 128 Byte, ein Menü, das sämtliche Funktionen umfaßt. Hier kann man den Drucker ein- und ausschalten, den nächsten oder vorigen Block laden, einen beliebigen anderen Block laden, denselben Block nochmal anzeigen und den Inhalt des Blocks im Speicher ändern.

Zum Ändern muß man zuerst die Bytenummer (in Hex) eingeben und dann den neuen Inhalt (ebenfalls in Hex). Gibt man als Adresse »X« ein, so wird das Ändern abgeschlossen, der geänderte Block wird ausgegeben und man kann entscheiden, ob der Block nun auf Diskette geschrieben werden soll.

Um einen beliebigen Block zu laden, muß man die Track- und Sektornummer (Dezimal) eingeben.

Der physikalische Aufbau der Blocks ist aus dem Manual des Laufwerks zu entnehmen. Man muß jedoch beachten, daß der Inhalt in Hex angezeigt und geändert wird und die Angabe der Block- und Sektorennummern dezimal erfolgt.

Die Bedienung des Programms erklärt sich durch die Menü-Steuerung von selbst. (Horst Wibbing/rg)

```
100 PRINT"▒▒▒▒"
101 REM COPYRIGHT (C) 1984 BY
102 REM HORST WIBBING, 62 WIESBADEN
110 PRINT"     ***************************
**"
120 PRINT"     *
 *"
130 PRINT"     *        DISK - DUMP  V2
 *"
140 PRINT"     *
 *"
150 PRINT"     ***************************
**"
151 PRINT"▒▒"
153 PRINT"IST DIE DISKETTE DRIN ? ";
154 GET Y$:IF(Y$<>"Y"ANDY$<>"J")THEN154
155 PRINT"▒"
160 REM ***** INIT *****
161 D=0:D$(0)="▒OFF▒":D$(1)="▒ON ▒"
162 TR=18:SE=0
163 PRINT"▒"
164 DIMB$(255)
165 GOTO270
166 REM ***** TRACK/SECTOR *****
170 PRINT"    WHICH TRACK ( 1 - 35 )";
180 INPUTTR
190 IFTR<1ORTR>35THENPRINT"▒▒":GOTO170
200 IFTR<18THENSR=20:GOTO240
210 IFTR<25THENSR=18:GOTO240
220 :IFTR<31THENSR=17:GOTO240
230 :SR=16
235 REM *****
240 PRINT"▒    WHICH SECTOR ( 0 -";SR;")
";
250 INPUTSE
260 IFSE<0ORSE>SRTHENPRINT"▒▒▒":GOTO240
265 REM *****
270 GOSUB1000:REM LESEN
280 GOSUB2000:REM AUSDRUCKEN
310 REM *****
320 PRINT"  1 => NEXT BLOCK       2 => SA
ME BLOCK"
330 PRINT"  3 => PREV. BLOCK      4 => OT
HER BLOCK"
340 PRINT"  5 => AENDERN          6 => PR
INTER ";D$(D)
350 PRINT"  0 => END"
360 GETT$:IFT$=""THEN360
370 IFT$="1"THENGOSUB3000:GOTO270
380 IFT$="2"THENGOTO280
385 IFT$="3"THENGOSUB3500:GOTO270
390 IFT$="4"THENGOTO170
391 IFT$="5"THENGOSUB5000:PRINT"▒":GOTO
320
392 IFT$="6"THENGOSUB2500:GOTO410
400 IFT$="0"THENGOTO9999
410 PRINT"▒▒▒▒":GOTO320
430 REM ***************** ENDE ******
1000 REM *******************
1001 REM **** SUB LESEN ****
1002 REM *******************
1010 OPEN15,8,15
1020 OPEN8,8,8,"#"
1025 PRINT#15,"U1 8 0"TR,SE
1030 FORI=0TO255
1040 :GET#8,B$(I)
1070 NEXTI
1080 CLOSE8:CLOSE15
1090 RETURN
1100 REM *********** ENDE SUB LESEN ****
*
2000 REM **********************
2001 REM **** SUB AUSDRUCK ****
2002 REM **********************
2010 IFTR<18THENBL=21*(TR-1)+SE:GOTO2050
2020 IFTR<25THENBL=19*(TR-1)+SE+34:GOTO2
050
2030 IFTR<31THENBL=18*(TR-1)+SE+34+24:GO
TO2050
2040 BL=17*(TR-1)+SE+34+24+30
2050 IFD=1THENOPEN4,4
2051 PRINT"▒▒ TRACK:";TR;"  SECTOR:";SE;
"  BLOCK:";BL"▒"
2055 IFD=1THENPRINT#4,"▒ TRACK:";TR;"  S
ECTOR:";SE;"  BLOCK:";BL"▒▒"
2060 PRINT"▒"
2070 FORI=0TO15
2071 :C$=""
```

Listing »Disk-Dump«

```
2072 :X=8*I:GOSUB 6000:REM DEC-HEX
2075 :PRINT" ";X$;" : ";
2076 :IFD=1THENPRINT#4," ";X$;" : ";
2080 :FORJ=0TO7
2088 :X=(ASC(B$(I*8+J)+CHR$(0)))
2089 :GOSUB 6000:REM DEC-HEX
2090 :PRINT" ";X$;
2091 :IFD=1THENPRINT#4," ";X$;
2095 :GOSUB4000:REM CHECK
2100 :NEXTJ
2110 :PRINTC$
2111 :IFD=1THENPRINT#4,C$
2120 NEXTI
2125 REM ***** 2.SEITE
2130 PRINT"▒   TASTE"
2140 GETT$:IFT$=""THEN2140
2150 PRINT"▒▒ TRACK:";TR;"  SECTOR:";SE;
"  BLOCK:";BL"▒"
2155 IFD=1THENPRINT#4,"▒▒ TRACK:";TR;"
SECTOR:";SE;"  BLOCK:";BL"▒▒"
2160 PRINT"▒"
2170 FORI=16TO31
2171 :C$=""
2172 :X=(8*I):GOSUB 6000:REM DEC-HEX
2175 :PRINT" ";X$;" : ";
2176 :IFD=1THENPRINT#4," ";X$;" : ";
2180 :FORJ=0TO7
2188 :X=(ASC(B$(I*8+J)+CHR$(0)))
2189 :GOSUB 6000:REM DEC-HEX
2190 :PRINT" ";X$;
2191 :IFD=1THENPRINT#4," ";X$;
2195 :GOSUB4000:REM CHECK
2200 :NEXTJ
2210 :PRINTC$
2211 :IFD=1THENPRINT#4,C$
2220 NEXTI
2225 IFD=1THENCLOSE4
2230 RETURN
2240 REM ***********ENDE SUB AUSDRUCK***
**
2500 REM *********************
2510 REM **** SUB PRINTER ****
2520 REM *********************
2530 IFD=0THEND=1:RETURN
2540 IFD=1THEND=0:RETURN
2550 REM *********** ENDE SUB PRINTER **
***
3000 REM ************************
3001 REM **** SUB NEXT BLOCK ****
3002 REM ************************
3010 SE=SE+1
3020 IFTR<18ANDSE>20THENTR=TR+1:SE=0:RET
URN
3030 IFTR<25ANDSE>18THENTR=TR+1:SE=0:RET
URN
3040 IFTR<31ANDSE>17THENTR=TR+1:SE=0:RET
URN
3050 IFTR<36ANDSE>16THENTR=TR+1:SE=0
3060 IFTR=36THENTR=0
3070 RETURN
3080 REM *********** ENDE SUB NEXT BLOCK
 *****
3500 REM *************************
3510 REM **** SUB PREV. BLOCK ****
3520 REM *************************
3530 SE=SE-1
3540 IFSE<0THENGOTO3600
3550 RETURN
3600 TR=TR-1
3610 SE=16
3620 IFTR=0THENTR=35:RETURN
3630 IFTR<18THENSE=20:RETURN
```

Listing »Disk-Dump«

```
3640 IFTR<25THENSE=18:RETURN
3650 IFTR<31THENSE=17
3660 RETURN
3670 REM *********** ENDE SUB PREV. BLOC
K *****
4000 REM ******************
4001 REM **** SUB CHECK ****
4002 REM ******************
4010 B=ASC(B$(I*8+J)+CHR$(0))
4020 IF((B<32)OR((B>127)AND(B<160)))THEN
C$=C$+" ":GOTO4030
4025 C$=C$+B$(I*8+J)
4030 RETURN
4040 REM *********** ENDE SUB CHECK ****
*
5000 REM ********************
5001 REM **** SUB AENDERN ****
5002 REM ********************
5010 PRINT"ADDR (00-FF/X=EXIT)";:INPUTAD
R$
5015 IFADR$=""THENGOTO5010
5020 IFADR$="X"THENGOTO5150
5030 IFLEN(ADR$)<2THENADR$="0"+ADR$
5040 IFLEN(ADR$)>2THENADR$=RIGHT$(ADR$,2
)
5045 X$=ADR$
5046 GOSUB 6500:REM HEX-DEC
5050 I=X
5060 PRINT"WERT (00-FF)";:INPUTW$
5065 IFW$=""THENGOTO5060
5070 IFLEN(W$)<2THENW$="0"+W$
5080 IFLEN(W$)>2THENW$=RIGHT$(W$,2)
5085 X$=W$
5086 GOSUB 6500:REM HEX-DEC
5090 B$(I)=CHR$(X)
5100 GOTO5000
5110 REM *****
5150 GOSUB2000:REM AUSDRUCKEN
5160 PRINT"AUF DISK SCHREIBEN";:INPUTY$
5170 IF((Y$<>"J")AND(Y$<>"Y"))THENGOTO53
00
5175 B1=ASC(B$(0)+CHR$(0))+1
5180 OPEN15,8,15
5190 OPEN8,8,8,"#"
5200 FORI=1TO255
5205 :IFB$(I)=""THENPRINT#8,CHR$(0);:GOT
O5220
5210 :PRINT#8,B$(I);
5220 NEXTI
5225 PRINT#15,"B-P:8",B1
5230 PRINT#15,"B-W:8,0"TR,SE
5240 CLOSE8:CLOSE15
5300 PRINT"WEITER AENDERN";:INPUTY$
5310 IF((Y$="J")OR(Y$="Y"))THENGOTO5000
5320 RETURN
5330 REM *********** ENDE SUB AENDERN **
***
6000 REM ********************
6010 REM **** SUB DEC-HEX ****
6020 REM ********************
6030 X1=0:X2=0
6040 X1=INT(X/16)
6050 X2=((X/16)-INT(X/16))*16
6100 IFX1<10THENX$=CHR$(X1+48):GOTO6120
6110 X$=CHR$(X1+55)
6120 IFX2<10THENX$=X$+CHR$(X2+48):RETURN
6130 X$=X$+CHR$(X2+55)
6140 RETURN
6150 REM *********** END SUB DEC-HEX ***
**
6500 REM ********************
6510 REM **** SUB HEX-DEC ****
```

```
6520 REM *********************
6530 X1$=LEFT$(X$,1)
6540 X2$=RIGHT$(X$,1)
6550 IF ASC(X1$)>64 THEN X=((ASC(X1$)-55
)*16):GOTO 6570
6560 X=((ASC(X1$)-48)*16)
6570 IF ASC(X2$)>64 THEN X=X+(ASC(X2$)-5
5):RETURN
6580 X=X+(ASC(X2$)-48)
6590 RETURN
6600 REM *********** END SUB HEX-DEC ***
**
9999 END
READY.
```

Listing »Disk-Dump« (Schluß)

Disketten-Organisation

Dieses Programm ermöglicht es, alle wichtigen Diskettenoperationen komfortabel und schnell durchzuführen.

Das Programm (Listing 1) ist für den VC 20 geschrieben, mit kleinen Änderungen jedoch auch auf dem C 64 lauffähig. Die für den C 64 notwendigen Änderungen beschränken sich im Wesentlichen auf die Anpassung des Bildschirmlayouts. Die im Listing vorkommenden Adressen 828 und 36879 sind für den C 64 durch 49152 und 53280/81 zu ersetzen.

Nach dem Starten erscheint sofort das Menü (Bild 1) auf dem Bildschirm. Durch Drücken der Tasten »+« oder »—« kann nun die Gerätenummer, unter der das Floppy-Laufwerk angesprochen wird, geändert werden. Diese Funktion ist sinnvoll, wenn mit zwei oder mehr Disketten-Stationen gearbeitet wird.

Über das Menü sind unter anderem alle Diskettenbefehle ausführbar, die sonst umständlich über den Kommandokanal gesendet werden müssen. Diese Befehle sind im Handbuch zur 1541-Floppy ausführlich erklärt, so daß an dieser Stelle nicht näher darauf eingegangen werden muß. Einige der Floppy-Kommandos stehen in erweiterter Form zur Verfügung. So kann zum Beispiel die Directory-Ausgabe wahlweise auf den Bildschirm oder über den Drucker erfolgen. Im letzteren Fall wird das Directory alphabetisch sortiert (Bild 2). Wird dies nicht gewünscht, dann muß in Zeile 1370 das »GOTO 1490« durch »RETURN« ersetzt werden.

Zusätzlich zu den über OPEN 1,8,15,... ansprechbaren Disketten-Befehlen können über das Menü noch einige weitere nützliche Funktionen aufgerufen werden. Unter anderem ist es möglich, Namen und ID einer Diskette oder die Gerätenummer der Floppy-Station zu ändern.

Außerdem sind Routinen zum Regenerieren und Überprüfen einer Diskette vorhanden. Weiterhin gibt es die Möglichkeit, die BAM einer Diskette alphabetisch zu sortieren. Diese Funktion erzeugt zwar ein übersichtliches, sortiertes Directory, sollte aber nur mit Vorsicht verwendet werden. Falls die BAM noch Einträge gelöschter Files enthält, kann es nach Abschluß des Sortiervorganges zu »Disk-Salat« kommen, da die Sortierroutine gelöschte Files nicht erkennt. Vor dem Einsatz dieser Funktion sollte die Diskette daher unbedingt mit dem Menüpunkt »Regenerieren« auf gelöschte Files überprüft werden.

Das Programm erklärt sich im übrigen weitgehend von selbst und stellt infolge der leichten und übersichtlichen Bedienbarkeit eine wirkliche Hilfe für den geplagten Disketten-Benutzer dar.

(Wilhelm Boosz/ev)

```
        1540TEST/DEMO      ZZ 2A
  4     DIR                PRG
  6     VIEW BAM           PRG
 14     DISPLAY T&S        PRG
  4     CHECK DISK         PRG
  9     PERFORMANCE TEST   PRG
  5     SEQUENTIAL FILE    PRG
 13     RANDOM FILE        PRG
        FREIE BLOECKE: 609
```

Bild 2. Sortiertes Directory bei Druckerausgabe

```
DISKETTEN-ORGANISATION
(c) by wilhelm boosz
floppy =        8 ■+/-■
■ nitialisieren
■ ORMATIEREN
■ ALIDIEREN
■ IRECTORY
■ OPIEREN
■ AM ALPHABETISCH
■ ERAETENR.AENDERN
■ MBENENNEN
■ OESCHEN
■ EGENERIEREN
■ AME DISK AENDERN
■ ENDERN DISK-ID
■ HECK DISK
■ NDE PROGRAMM
bitte waehlen !
```

Bild 1. Hardcopy des Bildschirm-Menüs

```
1 REM ***********************************
2 REM *DISKETTEN-ORGANISATIONS-PROGRAMM*
3 REM *        FUER DEN VC - 20         *
4 REM *          COPYRIGHT BY           *
5 REM *WILHELM BOOSZ - HEINRICHSTR. 21 *
6 REM *8602 STRULLENDORF - T.09543/7606*
7 REM ***********************************
8 REM
9 POKE828,8:POKE36879,25
10 GN%=PEEK(828):UR%=0
11 PRINT"■■DISKETTEN-ORGANISATION"
12 PRINT"■■(C) BY WILHELM BOOSZ"
13 PRINT"■■FLOPPY =     ■■■■■"GN%"■(+/-)"
14 PRINT"■■■I■ NITIALISIEREN":PRINT"■F■
ORMATIEREN"
15 PRINT"■V■ ALIDIEREN":PRINT"■D■ IRECTO
RY"
16 PRINT"■K■ OPIEREN":PRINT"■B■ AM ALPHA
BETISCH"
17 PRINT"■G■ ERAETENR.AENDERN":PRINT"■U■
 MBENENNEN"
18 PRINT"■L■ OESCHEN":PRINT"■R■ EGENERIE
REN"
19 PRINT"■N■ AME DISK AENDERN":PRINT"■A■
 ENDERN DISK-ID"
20 PRINT"■C■ HECK DISK":PRINT"■E■ NDE PR
OGRAMM"
21 PRINT"■■BITTE WAEHLEN !■"
22 GETY$:IFY$=""THEN22
23 IFY$="+"THENGN%=GN%+1:IFGN%>255THENGN
%=255
24 IFY$="-"THENGN%=GN%-1:IFGN%<0THENGN%=
0
25 IFY$="+"ORY$="-"THENPOKE828,GN%:PRINT
"■";:GOTO11
```

Listing 1. »Disketten-Organisation«

```
26 IFY$="E"THENPOKE36879,27:PRINT"":CL
R:END
27 IFY$="I"THENGOSUB44:CLR:GOTO10
28 IFY$="F"THENGOSUB50:CLR:GOTO10
29 IFY$="D"THENGOSUB64:CLR:GOTO10
30 IFY$="B"THENGOSUB84:CLR:GOTO10
31 IFY$="K"THENGOSUB121:CLR:GOTO10
32 IFY$="U"THENGOSUB130:CLR:GOTO10
33 IFY$="L"THENGOSUB148:CLR:GOTO10
34 IFY$="R"THENGOSUB159:CLR:GOTO10
35 IFY$="N"THENGOSUB212:CLR:GOTO10
36 IFY$="A"THENGOSUB221:CLR:GOTO10
37 IFY$="G"THENGOSUB231:CLR:GOTO10
38 IFY$="C"THENGOSUB263:CLR:GOTO10
39 IFY$="V"THENGOSUB58:CLR:GOTO10
40 GOTO22
41 REM ******************************
42 REM *I N I T I A L I S I E R E N *
43 REM ******************************
44 PRINT"       INITIALISIEREN"
45 CLOSE1:OPEN1,GN%,15,"I0":GOSUB298:CLO
SE1:IFEN=0THENRETURN
46 PRINT"   "EN;EN$;NE;EE:GOSUB305:RETUR
N
47 REM ******************************
48 REM *    F O R M A T I E R E N    *
49 REM ******************************
50 PRINT"       FORMATIEREN"
51 NA$="TEST":INPUT"    DISKETTENNAME";N
A$:NA$=LEFT$(NA$,16)
52 ID$="":INPUT"  DISKETTEN-ID";ID$
53 CLOSE1:OPEN1,GN%,15,"N:"+NA$+","+ID$:
CLOSE1:GOSUB298:IFENTHEN46
54 RETURN
55 REM ******************************
56 REM *    V A L I D I E R E N    *
57 REM ******************************
58 PRINT"       VALIDIEREN"
59 CLOSE1:OPEN1,GN%,15,"V":GOSUB298:CLOS
E1:IFEN=0THENRETURN
60 GOTO46
61 REM ******************************
62 REM *DIRECTORY DRUCKER/BILDSCHIRM*
63 REM ******************************
64 PRINT"    D I R E C T O R Y":PRINT"
D RUCKER"
65 PRINT"   B ILDSCHIRM"
66 GETY$:IFY$<>"D"ANDY$<>"B"THEN66
67 IFY$="D"THENPRINT"    DRUCKER BEREIT
?":GOTO69
68 GOSUB309:GOTO77
69 GETY$:IFY$<>"J"ANDY$<>"N"THEN69
70 IFY$="N"THENRETURN
71 PRINT"   GERAETENUMMER DRUCKER":PRINT
"    4    ";:INPUTGD%:STOP: OPEN4,GD%:GOSU
B309
72 PRINT#4,"        "F$(0)" "ID$:PRINT#4
73 FORY=1TOZE-1
74 IFLEN(F$(Y))<16THENF$(Y)=F$(Y)+" ":GO
TO74
75 PRINT#4,L$(Y)"    "F$(Y)"    "K$(Y)
76 NEXT:PRINT#4:PRINT#4,"        FREIE BLO
ECKE: "L$(ZE):PRINT#4:RETURN
77 PRINT"  NAME  "F$(0):PRINT"ID     "ID$
"  "
78 YY=3:FORY=1TOZE-1:YY=YY+1
79 PRINT" "L$(Y)" "F$(Y)" "TAB(19)LEFT$(
K$(Y),2):IFYY/18=INT(YY/18)THENGOSUB305:
PRINT""
80 NEXT:PRINT"  FREIE BLOECKE  "L$(ZE):G
OSUB305:RETURN
81 REM ******************************
82 REM *     BAM ALPHABETISCH       *
```

```
83 REM ******************************
84 DIMPN$(224),SP(29),SE(29):OPEN15,GN%,
15,"I"
85 NN=0:SN=0:SO=1:SP%=18:SE%=1
86 UR%=1:GOSUB299:IFENTHENCLOSE15:RETURN
87 OPEN2,GN%,2,"#":GOSUB299:IFENTHENCLOS
E15:GOTO46
88 PRINT""
89 PRINT#15,"U1";2;0;SP%;SE%:SP(SN)=SP%:
SE(SN)=SE%
90 GET#2,EG$:IFEG$=""THENEG$=CHR$(0)
91 SP%=ASC(EG$)
92 GET#2,EG$:IFEG$=""THENEG$=CHR$(0)
93 SE%=ASC(EG$):PZ=2
94 PRINT#15,"B-P";2;PZ:NN=NN+1
95 FORI=1TO30:GET#2,EG$:IFEG$=""THENEG$=
CHR$(0)
96 PN$(NN)=PN$(NN)+EG$:NEXTI
97 IFASC(LEFT$(PN$(NN),1))=0THENPN$="":L
O=LO+1:GOTO101
98 FORI=19TO5STEP-1:IFMID$(PN$(NN),I,1)<
>CHR$(160)THENI=5:GOTO100
99 PN$(NN)=LEFT$(PN$(NN),I-1)+CHR$(32)+R
IGHT$(PN$(NN),30-I)
100 NEXTI:PRINTMID$(PN$(NN),4,16)
101 IFPZ-226THENPZ=PZ+32:GOTO94
102 IFSP%THENPZ=2:SN=SN+1:GOTO89
103 IFLO=0THEN108
104 N$="":FORJ=1TO30:N$=N$+CHR$(0):NEXTJ
105 FORJ=1TOLO
106 PN$(NN+J)=N$
107 NEXTJ
108 PRINT"   FILES WERDEN SORTIERT! "
109 FORJ=1TONN-1:FORI=J+1TONN:IFMID$(PN$
(I),4,16)>MID$(PN$(J),4,16)THEN111
110 N$=PN$(I):PN$(I)=PN$(J):PN$(J)=N$:SO
=0
111 NEXTI,J:FORJ=1TONN:PRINTMID$(PN$(J),
4,16):FORI=19TO5STEP-1
112 IFMID$(PN$(J),I,1)<>CHR$(32)THENI=5:
GOTO114
113 PN$(J)=LEFT$(PN$(J),I-1)+CHR$(160)+R
IGHT$(PN$(J),30-I)
114 NEXTI,J:IFSOTHENCLOSE2:CLOSE15:RETUR
N
115 PN%=1:FORJ=0TOSN:PRINT#15,"B-P";2;0:
PRINT#2,CHR$(SP(J+1));CHR$(SE(J+1))
116 PZ=2:FORI=1TO8:PRINT#15,"B-P";2;PZ:P
RINT#2,PN$(PN%);:PN%=PN%+1:PZ=PZ+32
117 NEXTI:PRINT#15,"U2";2;0;SP(J);SE(J):
NEXTJ:CLOSE2:CLOSE15:RETURN
118 REM ******************************
119 REM *      K O P I E R E N       *
120 REM ******************************
121 PRINT"     K O P I E R E N"
122 PRINT"   WELCHES FILE SOLL":PRINT" K
OPIERT WERDEN?"
123 INPUT" ALTER NAME";AN$
124 INPUT" NEUER NAME";NN$
125 CLOSE1:OPEN1,GN%,15,"C:"+NN$+"="+AN$
:GOSUB298:IFENTHEN46
126 RETURN
127 REM ******************************
128 REM *    U M B E N E N N E N      *
129 REM ******************************
130 PRINT"     UMBENENNEN"
131 PRINT"  1  =  EIN FILE":PRINT" 2  =
 ALLE FILES"
132 PRINT"  BITTE WAEHLEN !"
133 GETY$:IFY$<"1"ORY$>"2"THEN133
134 IFY$="2"THEN139
135 INPUT"  ALTER NAME";AN$
136 INPUT" NEUER NAME";NN$
```

```
137 CLOSE1:OPEN1,GN%,15,"R:"+NN$+"="+AN$
:GOSUB298:IFENTHEN46
138 RETURN
139 GOSUB309
140 PRINT"":FORY=1TOZE-1:PRINT"NEUER NA
ME    "F$(Y):PRINT"";:INPUTNN
$
141 IFNN$=F$(Y)THEN144
142 CLOSE1:OPEN1,GN%,15,"R:"+NN$+"="+F$(
Y):GOSUB298
143 IFENTHENPRINT"     "EN;EN$;NE;EE:GOS
UB305
144 NEXT:RETURN
145 REM ******************************
146 REM *      L O E S C H E N       *
147 REM ******************************
148 PRINT"  LOESCHEN"
149 INPUT"FILENAME";NA$
150 PRINT"SOLL "NA$:PRINT"WIRKLICH WE
G ???"
151 GETY$:IFY$=""THEN151
152 IFY$<>"J"THENCLR:GOTO10
153 CLOSE1:OPEN1,GN%,15,"S0:"+NA$:GOSUB2
98:IFEN$<>"FILES SCRATCHED"THEN46
154 IFNE=0THENPRINT"KEIN FILE "NA$:PR
INT"GEFUNDEN !":GOSUB305:RETURN
155 PRINT""NE"FILE(S) GELOESCHT":GOSUB
305:RETURN
156 REM ******************************
157 REM *  R E G E N E R I E R E N   *
158 REM ******************************
159 GOSUB206:RE%=0:PRINT"   REGENERIERE
N"
160 SP%=1:TR%=DI%:OPEN15,8,15,"I0
161 OPEN8,8,2,"#2":FI%=0
162 PRINT#15,"U1:"2;0;TR%;SP%:GOSUB204
163 FORI=0TO1:GET#8,A$:GOSUB201:P%(I)=AS
C(A$):NEXT:A=2:GOSUB203
164 GET#8,A$:GOSUB201:A=ASC(A$)AND7
165 H$="":FORI=0TO1:GET#8,A$:GOSUB201:L%
(I)=ASC(A$):NEXT
166 FORI=3TO18:GET#8,A$:H$=H$+A$:NEXT:
167 IFH$=""THEN169
168 PRINT KI$(A);" ";CO$;H$;TAB(11)CO$;;
L%(0);L%(1):IFA=0THENGOTO185
169 FI%=FI%+1:IFFI%<8THENA=FI%*32+2:GOSU
B203:GOTO164
170 IFP%(0)=0THEN172
171 TR%=P%(0):SP%=P%(1):FI%=0:GOTO162
172 CLOSE8:IFRE%THEN58
173 GOSUB204:CLOSE15:RETURN
174 PRINT#15,"U1:"2;0;L%(0);L%(1):RETURN
175 D%=2:PRINT"SPUR, SEKTOR";
176 GOSUB174:FORI=0TO1:GET#8,A$:GOSUB201
:Z%(I)=ASC(A$)
177 PRINTZ%(I)" ";:NEXT:PRINT:PRINT"SP
UR, SEKTOR";
178 IFZ%(0)=0THENPRINT:RETURN
179 PRINT#15,"B-F"0;Z%(0);Z%(1):L%(0)=Z%
(0):L%(1)=Z%(1):Z%(D%)=Z%(0)
180 Z%(D%+1)=Z%(1):D%=D%+2
181 IFDS=0THEN176
182 PRINT:PRINT"DAS FILE WURDE BEREITS
UEBERSCHRIEBEN!":X%=NOT(X%):A=FI%*32+2
183 FORI=2TOD%-4STEP2:PRINT#15,"B-F:"0;Z
%(I);Z%(I+1):NEXT
184 L%(0)=TR%:L%(1)=SP%:GOSUB174:GOTO203
185 PRINT"ZURUECKHOLEN? (J/N)
186 GET A$:IF A$="" THEN 186
187 IF A$<>"J"THEN169
188 X%=0:PRINT"BLOECKE WIEDERBELEGEN...
":Z%(0)=L%(0):GOSUB175:IFX%THEN169
189 PRINT"PROGRAMM, SEQUENIELLES, 
USER,"
190 PRINT"ODER RELATIVES FILE ?"
191 GET JA$:IFJA$<>"P"ANDJA$<>"S"ANDJA$<
>"U"ANDJA$<>"R"THEN191
192 PRINT:IFJA$="P"THENKI%=130:GOTO196
193 IFJA$="U"THENKI%=131:GOTO196
194 IFJA$="R"THENKI%=132:GOTO196
195 KI%=129
196 PRINT"FILE ZURUECKHOLEN..."
197 L%(0)=TR%:L%(1)=SP%:GOSUB174
198 A=FI%*32+2:GOSUB203:PRINT#8,CHR$(KI%
);
199 BL%=((D%-2)/2)+1:PRINT""BL%"BLOECKE
";
200 PRINT#15,"U2:"2;0;TR%;SP%:PRINT"WIED
ERHERGESTELLT !":RE%=1:GOTO169
201 IFA$=""THENA$=CHR$(0)
202 RETURN
203 PRINT#15,"B-P:"2;A
204 INPUT#15,EN%,EM$,ET%,ES%:IFEN%=0THEN
RETURN
205 GOTO46
206 DIMP%(1),L%(1),Z%(300),KI$(4):CO$=CH
R$(34)
207 FORI=0TO4:READKI$(I):NEXT:DI%=18:RET
URN
208 DATA"DEL","SEQ","PRG","USR","REL"
209 REM ******************************
210 REM *     DISKNAME AENDERN       *
211 REM ******************************
212 OPEN15,GN%,15:NR=0:OPEN1,GN%,3,"#":P
RINT#15,"U1:3"NR",18,0"
213 PRINT#15,"B-P:3,144":D$="":FORL=1TO1
6:GET#1,A$:D$=D$+A$:NEXT
214 PRINT"ALTER DISK-NAME: "D$
215 NA$="DISK-NAME":PRINT"NEUER DISK-N
AME";:INPUTNA$
216 NA$=LEFT$(NA$+"                ",16
)
217 PRINT#15,"B-P:3,144":PRINT#1,NA$;:PR
INT#15,"U2:3"NR",18,0":CLOSE1:CLOSE15:RE
TURN
218 REM ******************************
219 REM *     DISK - ID AENDERN      *
220 REM ******************************
221 DV=0:OPEN15,GN%,15:OPEN1,GN%,3,"#":P
RINT#15,"U1:3"DV",18,0"
222 PRINT#15,"B-P:3,162":GET#1,A$,B$,C$,
D$,E$:PRINT"ALTER DISK-ID: "A$;B$;C$;
223 PRINTD$;E$:INPUT"NEUER DISK-ID";I$
224 IFLEN(I$)<3THENI$=I$+" ":GOTO224
225 I$=LEFT$(I$+"2A",5)
226 PRINT#15,"B-P:3,162":PRINT#1,I$;:PRI
NT#15,"U2:3"DV",18,0
227 GOSUB45:RETURN
228 REM ******************************
229 REM *GERAETENUMMER DISK AENDERN *
230 REM ******************************
231 PRINT"GERAETENUMMER AENDERN"
232 PRINT"BITTE ALLE FLOPPYS":PRINT"A
USSCHALTEN AUSSER"
233 PRINT"DER, DEREN NUMMER GE-":PRINT"A
ENDERT WERDEN SOLL!"
234 PRINT"ALTE NUMMER  8";
235 INPUTOD:IFOD<8OROD>15THEN234
236 PRINT"NEUE NUMMER  9";
237 INPUTND:IFND<8ORND>15THEN236
238 GOSUB239:GOSUB256:RETURN
239 CLOSE15:OPEN15,OD,15
240 PRINT#15,"M-R"CHR$(255)CHR$(255):GET
#15,C$:C=ASC(C$+CHR$(0))
```

Listing 1. »Disketten-Organisation« (Fortsetzung)

```
241 IFSTTHEN259
242 IFC=254THENMT=119
243 IFC=226THENMT=50
244 IFC=213THENMT=12
245 IFC=242THENMT=12
246 IFC=198THEN248
247 RETURN
248 PRINT#15,"M-R"CHR$(234)CHR$(16):GET#
15,ZB$:ZB=ASC(ZB$+CHR$(0))
249 IFZB=0THENMT=12
250 IFZB=1THEN252
251 IFSTTHEN259
252 PRINT#15,"M-R"CHR$(172)CHR$(16):GET#
15,ZC$:ZC=ASC(ZC$+CHR$(0))
253 IFZC=1THENMT=12
254 IFZC=2THENMT=12
255 RETURN
256 CLOSE15:OPEN15,OD,15
257 PRINT#15,"M-W"CHR$(MT)CHR$(0)CHR$(2)
CHR$(ND+32)CHR$(ND+64)
258 RETURN
259 PRINT"▒▒ FEHLER AM GERAET !":GOSUB30
2:RETURN
260 REM ******************************
261 REM *    C H E C K   D I S K    *
262 REM ******************************
263 DIMT(100):DIMS(100)
264 PRINT"▒▒▒   CHECK DISK"
265 GOSUB59:OPEN15,GN%,15:N%=RND(TI)*255
266 A$="":FORI=1TO255:A$=A$+CHR$(255AND(
I+N%)):NEXT
267 UR%=1:GOSUB299:IFENTHENCLOSE15:GOTO4
6
268 OPEN2,GN%,2,"#"
269 PRINT:PRINT#2,A$;
270 T=1:S=0
271 PRINT#15,"B-A:0"T;S
272 INPUT#15,EN,EM$,ET,ES
273 IFEN=0THEN276
274 IFET=0THEN284
275 PRINT#15,"B-A:0"ET;ES:T=ET:S=ES
276 PRINT#15,"U2:2,0"T;S
277 NB=NB+1:PRINT"GETESTETE BLOECKE"NB
278 PRINT"▒ SPUR      ▒▒▒▒"T;" SEKTOR
▒▒▒▒"S"▒▒▒"
279 INPUT#15,EN,EM$,ES,ET
280 IFEN=0THEN271
281 T(J)=T:S(J)=S:J=J+1
282 PRINT"▒▒▒▒▒FEHLERHAFTER BLOCK:",T;S"
▒▒▒▒▒▒"
283 GOTO271
284 GOSUB45
285 GOSUB299:IFENTHENCLOSE15:CLOSE2:GOTO
46
286 CLOSE2:CLOSE15
287 IFJ=0THENPRINT"▒▒▒▒▒▒KEINE FEHLERHAF
TEN BLOECKE !":RETURN
288 OPEN2,GN%,2,"#"
289 PRINT"▒▒FEHLERBLOECKE","SPUR","SEKTO
R"
290 FORI=0TOJ-1
291 PRINT#15,"B-A:0"T(I);S(I)
292 PRINT,,T(I),S(I)
293 NEXT
294 PRINT"▒"J"FEHLERBLOECKE WURDEN FESTG
ESTELLT":CLOSE15:CLOSE2:GOSUB302:RETURN
295 REM ******************************
296 REM * FEHLERKANAL UEBERPRUEFEN  *
297 REM ******************************
298 CLOSE15:OPEN15,GN%,15
299 INPUT#15,EN,EN$,NE,EE:IFUR%THENRETUR
N
300 CLOSE15
301 RETURN
302 REM ******************************
303 REM *    TASTENDRUCK ABWARTEN   *
304 REM ******************************
305 PRINT"▒WEITER-->TASTE !":POKE198,0:W
AIT198,1:POKE198,0:RETURN
306 REM ******************************
307 REM *     DIRECTORY LESEN       *
308 REM ******************************
309 PRINT"▒▒▒▒▒   DIRECTORY LESEN":PRIN
T"▒▒▒▒▒MOMENT BITTE !"
310 DIML$(140),K$(140),F$(140)
311 OPEN1,GN%,0,"$0":OPEN15,GN%,15:UR%=1
:GOSUB299:IFENTHEN46
312 GET#1,A$,B$:ZE=-1
313 ZE=ZE+1:GET#1,A$,B$
314 GET#1,A$,B$:C=0
315 IFA$<>""THENC=ASC(A$)
316 IFB$<>""THENC=C+ASC(B$)*256
317 L$(ZE)=RIGHT$("    "+MID$(STR$(C),2),
3)
318 GET#1,B$:IFST<>0THENFB$=RIGHT$("    "
+L$,3):GOTO327
319 IFB$<>CHR$(34)THEN 318
320 F$(ZE)=""
321 GET#1,B$:IFB$<>CHR$(34)THENF$(ZE)=F$
(ZE)+B$:GOTO321
322 GET#1,B$:IFB$=CHR$(32)THEN322
323 C$=""
324 C$=C$+B$:GET#1,B$:IFB$<>""THEN324
325 K$(ZE)=LEFT$(C$,3):IFZE=0THENID$=LEF
T$(C$,5)
326 IFST=0THEN313
327 CLOSE1:CLOSE15:RETURN
READY.
```

Listing 1. »Disketten-Organisation« (Schluß)

Programmiertes LISTing: LISTX-Y

Bei Hilfsprogrammen, die viele Benutzeranleitungen enthalten, gibt man diese Anleitungen normalerweise über PRINT-Anweisungen auf dem Bildschirm aus. An sich würde es bei geschickter Formulierung jedoch reichen, die sowieso vorhandenen REM-Erläuterungen als Anleitungen für den Benutzer mitzuverwenden.

Nur, wie bringt man diese auf den Bildschirm? Mit LIST wird das Basic-Programm jedesmal zum Direktmodus hin verlassen und, was noch schwerer wiegt, die Anfangs- und Endzeilennummern können nur als direkte Zahlen, nicht über Variablen, angegeben werden. Unser Programmvorschlag (Listing 1) simuliert den im VC 20-Basic nicht vorhandenen Befehl LISTX-Y (X=Variable, die die Anfangszeilennummer des zu listenden Programms übergibt, Y = Endzeilennummer).

Man könnte sich ein solches Hilfsprogramm durch Beschreiben des Tastaturpuffers konstruieren. Dann würden sich aber die in den Puffer geschriebenen LIST-Anweisungen auf dem Bildschirm störend bemerkbar machen. Andererseits kommt aus Geschwindigkeitsgründen nur ein Maschinenprogramm in Frage. Will man ein solches Maschinenprogramm per DATA-Zeilen einlesbar gestalten, wäre der Aufwand recht hoch. Außerdem gäbe es Schwierigkeiten mit dem nur immer auf den Datenanfang zurückstellbaren DATA-Zeiger (kein RESTORE X vorhanden). Einlesen per POKE wäre noch aufwendiger. Umgekehrt muß aber ein solches Maschinenprogramm zur Simulation von LIST Y notwendigerweise viele Teile enthalten, die bereits im Betriebssystem vorkommen. Wir lösen das Problem, indem wir mit zwei einfachen FOR-NEXT-Schleifen geeignete Teile des Betriebssystems in den Kassettenpuffer kopieren und diese Kopien dann durch sechs POKE-Anweisungen so abändern, daß sie unseren Ansprüchen genügen. Listing 1 zeigt das entsprechende Unterprogramm, das beim ersten Aufruf mit »GOSUB 63000« die Maschinenroutine erzeugt. Alle weiteren Aufrufe können mit »GOSUB 6350« erfolgen, wodurch, einiges an Zeit gespart wird. In der Variablen X wird die Anfangszeile, in Y die Endzeile übergeben. Listing 3 zeigt ein Demo-Programm. In Listing 2 gebe wir ein Anwendungsbeispiel an: In einem längeren Programm mögen alle REM-Erläuterungen in den Zeilen 100*a bis 100*a+4 untergebracht sein, also in den ersten fünf Zeilen ab jeder vollen Hunderternummer. Das Beispielprogramm nach Listing 2 wird per GOSUB63100 angesprungen und listet per Cursor-Down-Taste alle oben genannten REM-Zeilen (und nur diese, eventuelle weitere REM-Zeilen werden nicht berücksichtigt). Und zwar geschieht dies in Endlosform, das heißt nach Durchgang durch die letzte zu listende REM-Erläuterung erscheint wieder die erste und so fort. Durch Drücken der Return-Taste gelangt man wieder ins Hauptprogramm.

(Fred Behringer/ev)

```
63000 REM LIST X-Y
63001 REM ========
63002 REM
63010 FORI=0TO45:POKE828+I,PEEK(50707+I)
:NEXT
63020 POKE845,252:POKE856,251
63030 FORI=0TO120:POKE874+I,PEEK(50889+I
):NEXT
63040 POKE896,254:POKE900,253:POKE935,19
:POKE949,96
63050 POKE252,X/256:POKE251,X-256*PEEK(2
52)
63060 POKE254,Y/256:POKE253,Y-256*PEEK(2
54)
63070 SYS828:RETURN
READY.
```

Listing 1. Programm zur Simulation von LISTX-Y. Ansprung per GOSUB63000. Einlesezeit 2,6 Sekunden. Jeder weitere Ansprung per GOSUB63050 mit »sofortiger« Abarbeitung.

60010	Einlesen von \$C613-\$C640 (50707-50752). Maschinenprogramm zur Berechnung der Startadresse einer Basic-Zeile. Zeilennummereingabe (der Anfangszeile) in \$0014/\$0015 (20/21).
63020	Verlegen dieser Parameterübergabestellen nach \$00FB/\$00FC (251/252), da \$0014/\$0015 (20/21) beim Ansprung SYS828 (siehe unten) gestört werden
63030	\$C6009-\$C741 (50889-51009). Programm zum Auflisten der gewünschten Zeilen. In diesem Programm wird die laufende Nummer der gerade gelisteten Zeile mit der beim Ansprung dieser Routine an \$0014/\$0015 (20/21) übergebenen Nummer der Endzeile verglichen, und die Routine wird verlassen, wenn die Endnummer erreicht ist. Auch hier läßt sich aus dem eben erwähnten Grund die Stelle \$0014/\$0015 (20/21) nicht als Parameterübergabestelle (von Basic zu Maschinenprogramm) verwenden und wird in 63040 zu \$00FD/\$00FE (253/254) abgeändert. Überschreiben des JMP-Befehls in \$C714 (50964) mit einem RTS-Befehl sorgt dafür, daß nicht zum Basic-Warmstart zurückgekehrt wird, sondern zum Hauptprogramm. Die Sprungadresse \$0306/\$0307 (774/775), die in \$C717 (50967) wirksam wird, hätte geändert werden müssen, da sich der Sprung ja nun auf die der Stelle \$C71A (50970) entsprechenden Stelle in der Kopie im Kassettenpuffer bezieht. Es war einfacher, den (hier) überflüssigen JMP-Befehl in \$C717 (50967) zu umgehen, indem die Sprungadresse in der der Stelle \$C705 (50949) entsprechenden Stelle der Kopie abgeändert wurde.
63050	Die vom Hauptprogramm in X stammende Nummer der Anfangszeile wird in Low Byte/High Byte gespalten und der Stelle 251/252 übergeben (siehe oben)
63060	Entsprechend für die aus Y stammende Nummer der Endzeile und 253/254
63070	Aufruf des Maschinenprogramms und Rückkehr ins Basic-Hauptprogramm. Nach Rückkehr steht in \$030D/\$030C (781/780) die Nummer (Low Byte/High Byte) der nächstanstehenden Basic-Zeile zur Verfügung.

Beschreibung des Programms nach Listing 1 zur Simulation von LISTX-Y

```
63100 REM AUSDRUCKEN DER ZEILEN 100*A BI
S 100*A+4
63105 REM
63110 GOSUB63000:PRINTCHR$(147);
63120 X=0
63130 GETX$:IFX$=""THEN63130
63140 IFX$=CHR$(13)THEN63200
63150 IFX$<>CHR$(17)THEN63120
63160 Y=X+4:GOSUB63050:PRINTCHR$(145);
63170 X=X+100:IFX>63999THENX=0:GOTO63160
63180 I=PEEK(781)+256*PEEK(780):IFI<Y+10
RI>X+4THENX=INT(I/100)*100
63190 GOTO63130
63200 RETURN
READY.
```

Listing 2. Programm zum Ausdrucken aller Zeilen mit den Nummern 100*a bis 100*a+4, wobei a von 0 bis 633 geht. Nicht vorhandene Zeilen werden ohne Verzögerung übersprungen. Der Ausdruck läuft, solange die Cursor-Down-Taste gedrückt ist, auch über 63304 hinaus (Wiederanfang bei 0).

63110	Einlesen des Simulationsprogramms für LISTX-Y nach Listing 1
63120	Anfangszeile auf Null
63130 63140	Wenn Return-Taste, dann zurück ins Hauptprogramm

63150	Wenn nicht Cursor-Down- oder Return-Taste, dann Anfang wieder 0.
63160	Wenn Cursor-Down-Taste, dann Listen der Zeilen X bis X+4 und Ausgleich für Zeilenvorschub
63170	Sprung um 100 nach vorn. Wenn Vorrat erschöpft, Zeilennummer wieder auf 0 (Endlosdurchlauf).
63180	Schnelles Vortasten zur nächsten vorhandenen Zeile mit Nummer zwischen 110*a und 100*a+4
63190	Weiter in Eingabeschleife

Beschreibung des Programms nach Listing 2 zum Ausdrucken aller Kopf-REM-Zeilen. Das Programm verwendet die Routine für LISTX-Y aus Listing 1.

```
10 REM LIST-DEMO
20 REM =========
30 REM
40 REM
50 REM
60 REM
70 PRINT"LIST-DEMO"
80 PRINT
90 INPUT"VON";X:INPUT"BIS";Y
100 PRINT
110 GOSUB63000:REM LIST X-Y
120 PRINT:PRINT"** OK ** (TASTE)"
130 GETA$:IFA$=""THEN130
140 PRINT:PRINT
150 GOTO70
160 REM
170 REM
180 REM
READY.
```

Listing 3. LIST-Demo zum Testen der Routine nach Listing 1

Kopieren mit Komfort: Super Copy

Bereits im ersten 64'er wurde Ihnen mit »Disk Copy« ein Kopierprogramm für Disketten vorgestellt. Wir möchten Ihnen heute eine völlig revidierte Fassung vorstellen, die erheblich leistungsfähiger und komfortabler ist.

Was soll ein gutes Kopierprogramm leisten? Nun, zunächst einmal soll es kopieren. Dazu muß man auswählen können, was kopiert werden soll. Dieser Vorgang dauert beim »Disk Copy« sehr lange, vor allem, wenn auf der Quelldiskette viele Programme sind. Bei »Super Copy« geht es genauso schnell wie das Einlesen eines Directory. Fehler beim Kopieren sollen möglichst ohne Programmabsturz und völligen Neubeginn behebbar sein. Gerade hier liegt eine Stärke von »Super Copy«. Alle Funktionen, die man sonst noch beim Kopieren braucht (Formatieren, Gültigkeitskontrolle (Vality check) etc.), sollen integriert sein. Dazu gehört auch eine komfortable Löschmöglichkeit, um Disketten »aufzuräumen«. Schließlich soll das Programm möglichst wenig Speicherplatz belegen, damit zum Kopieren genug zur Verfügung steht.

Aus all dem ergibt sich eine Konsequenz: Ein solches Programm läßt sich nur in Maschinensprache schreiben, da Basic einfach zu langsam ist und zuviel Speicherplatz belegt. Trotzdem habe ich einige Einschränkungen gemacht:

Relative Files können nicht kopiert werden. Dies verlangt eine zu aufwendige Verwaltung und kommt auch zu selten vor, um es ins Programm zu integrieren. In einem Programmdurchlauf können höchstens 32 Files kopiert werden. Mehr Filenamen kann das Programm nicht speichern.

Ansonsten aber läßt »Super Copy« kaum noch Wünsche offen, höchstens den nach einem schnelleren Laufwerk. Aber auch da läßt sich wohl noch etwas machen.

Wie arbeitet »Super Copy«?

Nach dem Programmstart meldet sich das Programm mit einem Menü:

1. Directory
2. Kopieren
3. Formatieren
4. Scratchen
5. Validieren
6. Ende

Durch Druck auf eine Ziffer wählen Sie die entsprechende Funktion aus. Übrigens können Sie im Programm immer dann, wenn Sie irgendeine Taste drücken müssen, mit »←« in dieses Menü zurückkehren. Gehen wir nun die einzelnen Funktionen einmal durch.

Zur Funktion »Directory« ist nicht viel zu sagen. Es erscheint das Verzeichnis aller Files auf der Diskette.

Beim »Formatieren« müssen Sie den Diskettennamen und die ID — wie üblich durch ein Komma getrennt — angeben. Eine ID ist nur bei einer neuen Diskette wichtig. Verzichten Sie darauf, werden zwar alle Einträge im Directory gelöscht, aber es entfällt das Neuformatieren der einzelnen Spuren. »Validieren« (Gültigkeitskontrolle) entspricht dem Basic-Befehl OPEN 1,8,15,"V":CLOSE 1.

Entscheiden Sie sich für »Kopieren«, werden Sie aufgefordert, die Quelldiskette einzulegen. Nach Tastendruck erscheinen nun die Namen der Programme. Files, die kopiert werden sollen, kennzeichnen Sie mit der »J«-Taste, die anderen mit »N«. Relative Files können nicht kopiert werden, daher erscheint eine Fehlermeldung, wenn Sie versuchen, solche Files mit »J« zu markieren. Das Programm kann maximal 32 Namen speichern. Wenn Sie mehr als 32 Files kopieren wollen, erscheint die Fehlermeldung »Kopierliste voll«. Sie können nun die bisher markierten Programme kopieren und nach Abschluß einen neuen Programmdurchlauf starten. Haben Sie Ihre Auswahl beendet, gibt das Programm an, wieviele Blöcke insgesamt zu kopieren sind, damit Sie genügend Platz auf der Zieldiskette bereitstellen können. Ein neues Menü erscheint:

1. Directory
2. Formatieren
3. Validieren

*** Space***
für weiter

Sie können nun in aller Ruhe eine Zieldiskette aussuchen, eventuell noch formatieren etc. Sie kommen in jedem Fall in dieses Menü zurück. Sind alle Vorbereitungen abgeschlossen, drücken Sie »Space«, um mit dem Kopieren fortzufahren. Das Programm fordert nun auf, die Quelldiskette einzulegen, und liest die vorher markierten Programme ein. Sollte dabei ein Fehler auftreten, weil Sie zum Beispiel aus Versehen die falsche Diskette eingelegt haben, wird eine entsprechende Meldung ausgegeben und gefragt, ob dieses File übersprungen oder ein neuer Versuch unternommen werden soll. Auch Lese-

fehler des Laufwerks werden in dieser Weise gehandhabt. In einem Durchgang können maximal 234 Blöcke eingelesen werden. Ist noch mehr zu kopieren wird das Einlesen abgebrochen.

Jetzt müssen Sie angeben, ob Sie fortlaufend oder einzeln kopieren möchten. Fortlaufend bedeutet, daß die Files der Reihe nach auf dieselbe Diskette geschrieben werden. Beim Einzelkopieren dagegen springt das Programm nach jedem Schreibvorgang wieder in ein Menü, und Sie haben die Möglichkeit, die Diskette zu wechseln, ein Directory anzusehen, zu formatieren oder zu validieren. Außerdem können Sie auch das zuletzt kopierte Programm noch einmal auf eine andere Diskette kopieren. Das jeweils nächste File wird vor dem Schreiben angezeigt, damit Sie die richtige Zieldiskette einlegen können.

Haben Sie Ihre Wahl getroffen, läuft der Schreibvorgang in der oben beschriebenen Art und Weise ab. Schreibfehler werden wie Lesefehler behandelt, das heißt es wird gefragt, ob ein neuer Versuch gestartet oder das File übersprungen werden soll.

Sind alle Programme kopiert, erscheint die Meldung »KOPIE FERTIG«. Sind aber nach dem ersten Lese/Schreib-Durchgang noch weitere Programme zu kopieren, fährt das Programm mit der Aufforderung zum Einlegen der Quelldiskette fort.

»Scratchen«, also das Löschen von Programmen, gehört zu den angenehmsten Funktionen, die »Super Copy« zu bieten hat. Seien Sie aber vorsichtig, sonst haben Sie bald überhaupt keine Programme mehr. Der Ablauf ist ähnlich wie beim Kopieren. Doch anstatt die Files zum Kopieren zu kennzeichnen, werden sie nun zum Löschen markiert. Sollten Sie einen Fehler gemacht haben, ist allerdings noch nichts verloren, denn am Ende des Markiervorgangs müssen Sie noch einmal ausdrücklich durch Drücken der »Space«-Taste bestätigen, daß es Ihnen ernst ist.

Zum Programmaufbau:

Ausführliche Erläuterungen des Assemblerquelltextes möchte ich Ihnen und mir ersparen. Nur soviel: Das Programm ist im wesentlichen modular aufgebaut, das heißt es besteht aus einzelnen Blöcken, die von den verschiedenen Menüs aus angesprungen werden. Daher ist es nicht nötig, häufig vorkommende Programmteile wie Tasten- oder Fehlerabfragen ständig neu zu schreiben. Auch der Teil zum Markieren der Programme ist beim »Scratchen« und beim Kopieren identisch. Ein Flag, also eine Speicherstelle, deren Inhalt etwas »signalisieren« soll, entscheidet darüber, welche Texte ausgegeben und wohin zurückgesprungen werden soll.

Eine solche Programmiertechnik trägt dazu bei, die Programme ebenso kurz wie übersichtlich zu halten.

Hinweise zum Abtippen:

Bevor Sie sich ans Eintippen des Ladeprogramms machen, müssen Sie unbedingt folgende Befehle im Direktmodus eingeben:

POKE 43,1: POKE 44,32: POKE 8192,0: NEW (RETURN)

Damit wird der Start des Basic-Speichers nach oben verlegt. Tun Sie das nicht, steht Ihr Basic-Ladeprogramm dort, wo nachher das fertige Kopierprogramm hin' gePOKEt' wird.

Ich weiß, daß es eine Zumutung ist, eine derartige Menge von Zahlen abzutippen, und kein Mensch wird dies auf Anhieb fehlerfrei bewerkstelligen können. Um Ihnen aber die Fehlersuche so leicht wie möglich zu machen, habe ich den 'DATA-Zeilenberg' in 11 Blöcke eingeteilt. Zu jedem dieser Blöcke gehört eine Prüfsumme. Stimmt etwas nicht, gibt das Ladeprogramm an, in welchem Block der Fehler steckt. Achten Sie aber vor allem darauf, daß Sie am Zeilenende kein Komma mehr schreiben, denn das interpretiert der Computer als Null!

Nach dem Abtippen sollten Sie vor dem ersten Programmlauf dieses unbedingt abspeichern, denn sonst könnte es passieren, daß sich das Programm durch eine fehlerhafte Eingabe selbst löscht, und die ganze Mühe war umsonst. Vor dem Laden des Programms müssen Sie die obengenannte Zeile eintippen, um die Basic-Startadresse wieder hochzulegen.

Starten Sie dann das Programm mit »RUN«. Erscheint während des Programmablaufs die Fehlermeldung »TYPE MISMATCH ERROR«, dann haben Sie bestimmt ein »l« statt einer »1« eingetippt oder ein »O« statt einer »0«. Ansonsten müssen Sie alle Fehler in den DATA-Blöcken korrigieren. Wenn alles richtig ist, meldet sich das Ladeprogramm nach einiger Zeit mit »READY«. Nun steht das Programm fertig im Speicher. Um es auf Diskette abzuspeichern, müssen jetzt folgende Befehle eingetippt werden:

SYS64738 (RETURN)

Keine Angst! Das ist zwar ein Befehl, der den Computer in den Zustand direkt nach dem Einschalten zurückversetzt, aber unser Programm ist nach wie vor im Speicher vorhanden. Nun noch folgende Zeilen:

10 SYS 2064 (RETURN)

Das ist der Startbefehl für »Super Copy«

POKE 45, 251 : POKE 46, 18

Damit wird das Basic-Ende auf das Programmende gesetzt. Jetzt können Sie mit

SAVE »Super Copy«, 8

das fertige Programm auf Diskette abspeichern.

(Dietrich Weineck/gk)

```
1 REM ****************************
2 REM *                          *
3 REM *     SUPER COPY 1541      *
4 REM * VON N.MANN & D.WEINECK *
5 REM *      FLEETRADE 40        *
6 REM *      2800 BREMEN         *
7 REM *   TEL. 0421 / 493090     *
8 REM *                          *
9 REM ****************************
10 FORI=0TO10:READA:PR(I)=A:NEXT
20 SA=2064:I=0
30 PA=SA+256*I:CH=0
40 FORJ=0TO255:READA:POKEPA+J,A:CH=CH+A:
NEXT
50 IFCH<>PR(I)THEN90
55 I=I+1:IFI<10THEN30
60 PA=PA+256:CH=0
70 FORJ=0TO234:READA:POKEPA+J,A:CH=CH+A:
NEXT
80 IFCH=PR(I)THENEND
90 PRINT"FEHLER IN BLOCK"I+1:END
100 DATA28785,26329,30148,25856,28858,29
119,27622,19508,15126,14152,13792
110 REM
120 REM *** BLOCK 1 ***
130 REM
140 DATA169,11,141,32,208,141,33,208,169
,1,141,134,2,169,139,160,15,32,11,15
150 DATA32,231,14,201,49,144,249,201,55,
176,245,32,210,255,201,49,208,6,32
160 DATA97,8,76,16,8,201,50,208,3,76,194
,8,201,51,208,6,32,230,8,76,16,8,201
170 DATA52,208,3,76,78,9,201,53,208,6,32
,185,9,76,16,8,76,116,164,32,68,229
180 DATA169,8,170,160,0,32,186,255,169,1
,162,237,160,18,32,189,255,32,192,255
190 DATA169,8,32,180,255,169,0,32,150,25
5,32,165,255,32,165,255,76,166,8,32
200 DATA165,255,133,251,32,165,255,166,2
51,32,205,189,32,237,14,32,165,255
210 DATA240,6,32,210,255,24,144,245,32,2
40,14,32,165,255,32,165,255,165,144
```

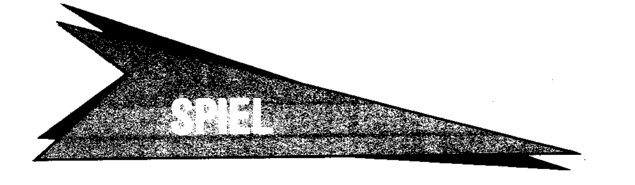
»Super Copy«

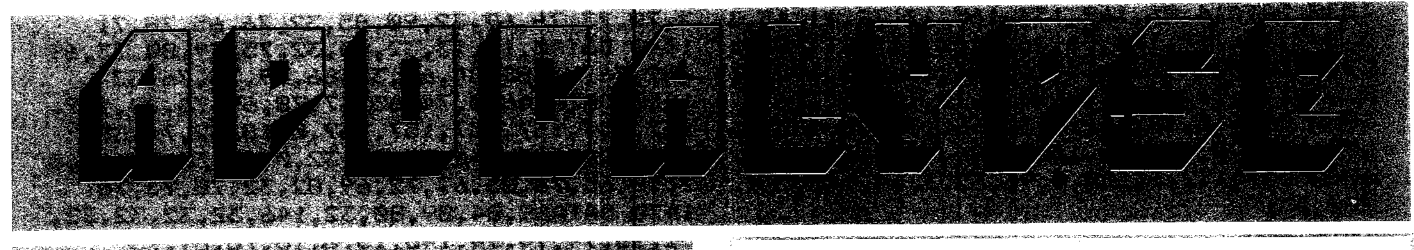
```
220 DATA240,216,32,171,255,169,8,32,195,
255,32,136,14,32,245,14,96,32,209,9
230 DATA169,111,160,18,32,11,15,169,255,
141,195,2,32,117,11,32,222,11,32,195
240 DATA11,32,117,11,32,234,12,32,227,11
,76,221,8,169,168,160,17,32,11,15,162
250 DATA0,32,207,255,157,64,3,232,201,13
,208,245,169,0,202,157,64,3,169,78
260 DATA141,61,3,169,48,141,62,3,169,58,
141,63,3,169
270 REM
280 REM *** BLOCK 2 ***
290 REM
300 DATA15,162,8,168,32,186,255,162,3,18
9,61,3,240,3,232,208,248,138,162,61
310 DATA160,3,32,189,255,32,192,255,32,1
63,14,8,169,15,32,195,255,40,176,1
320 DATA96,169,200,160,17,32,11,15,32,24
5,14,201,74,240,185,201,78,208,245
330 DATA76,16,8,32,68,229,169,32,160,18,
32,11,15,169,128,141,196,2,32,224,9
340 DATA169,8,32,195,255,173,193,2,208,3
,76,16,8,32,68,11,169,135,160,18,32
350 DATA11,15,32,222,11,169,83,141,61,3,
169,48,141,62,3,169,58,141,63,3,138
360 DATA24,105,3,162,61,160,3,32,189,255
,169,15,162,8,168,32,186,255,32,192
370 DATA255,169,15,32,195,255,174,192,2,
232,236,193,2,176,9,238,192,2,32,18
380 DATA12,76,122,9,76,16,8,169,86,141,2
39,18,169,27,160,17,32,11,15,32,75
390 DATA14,169,73,141,239,18,32,136,14,9
6,32,68,229,169,0,141,196,2,169,46
400 DATA160,17,32,11,15,32,245,14,32,240
,14,32,240,14,32,75,14,32,136,14,144
410 DATA5,104,104,76,16,8,169,8,170,160,
0,32,186,255,169,1,162,237,160,18,32
420 DATA189,255,32,192,255,169,8,32,180,
255,169
430 REM
440 REM *** BLOCK 3 ***
450 REM
460 DATA0,32,150,255,160,4,32,165,255,13
6,208,250,32,165,255,133,251,32,165
470 DATA255,166,251,32,205,189,32,237,14
,32,165,255,240,6,32,210,255,24,144
480 DATA245,32,240,14,32,240,14,32,165,2
55,32,165,255,160,0,140,193,2,32,165
490 DATA255,141,192,2,32,165,255,141,194
,2,174,192,2,32,205,189,32,237,14,160
500 DATA0,32,165,255,32,210,255,153,64,3
,240,3,200,208,242,32,165,255,32,165
510 DATA255,165,144,240,9,44,196,2,16,1,
96,76,50,11,173,194,2,208,7,173,192
520 DATA2,201,235,144,10,169,70,160,16,3
2,11,15,76,44,11,173,193,2,201,32,144
530 DATA10,169,86,160,16,32,11,15,24,144
,214,169,0,133,8,169,32,133,211,169
540 DATA110,160,16,32,11,15,32,245,14,20
1,78,240,104,201,74,208,245,169,125
550 DATA160,16,32,11,15,173,193,2,32,233
,13,234,234,162,0,232,189,64,3,201
560 DATA34,208,248,142,194,2,232,189,64,
3,201,34,240,7,145,65,234,232,200,208
570 DATA242,138,172,193,2,24,237,194,2,1
53,0,19,173,192,2,153,160,19,189,64
580 DATA3,208,15,44,196,2,48,21,169,135,
160,16,32,11,15,76,44
590 REM
600 REM *** BLOCK 4 ***
610 REM
620 DATA11,201,83,240,7,201,80,240,3,232
,208,225,153,32,19,238,193,2,24,144
630 DATA7,169,170,160,16,32,11,15,32,240
,14,76,72,10,32,171,255,169,8,32,195
640 DATA255,173,193,2,208,5,104,104,76,1
6,8,162,0,142,198,2,142,197,2,32,240
650 DATA14,32,240,14,169,18,32,210,255,2
4,189,160,19,109,197,2,141,197,2,169
660 DATA0,109,198,2,141,198,2,232,236,19
3,2,144,233,174,197,2,76,205,189,169
670 DATA234,160,16,32,11,15,44,195,2,48,
7,169,215,160,18,32,11,15,169,160,160
680 DATA18,32,11,15,32,245,14,201,32,208
,1,96,201,50,208,6,32,185,9,76,117
690 DATA11,201,49,208,6,32,97,8,76,117,1
1,201,51,208,6,32,230,8,76,117,11,201
700 DATA52,208,214,44,195,2,48,209,104,1
04,76,244,12,169,180,160,16,32,11,15
710 DATA32,245,14,201,49,240,7,201,50,20
8,245,169,0,44,169,255,141,195,2,96
720 DATA162,0,142,192,2,169,22,157,64,19
,169,0,141,194,2,44,196,2,16,10,169
730 DATA62,160,18,32,11,15,24,144,10,138
,240,29,169,46,160,17,32,11,15,32,245
740 DATA14,44,196,2,16,14,201,32
750 REM
760 REM *** BLOCK 5 ***
770 REM
780 DATA208,244,169,96,160,18,32,11,15,2
4,144,7,169,78,160,17,32,11,15,172
790 DATA192,2,190,0,19,173,192,2,32,233,
13,177,65,32,210,255,200,202,208,247
800 DATA32,28,15,174,192,2,189,0,19,133,
251,173,192,2,32,233,13,162,0,177,65
810 DATA157,64,3,200,232,198,251,208,245
,44,196,2,16,1,96,160,0,185,241,18
820 DATA157,64,3,200,232,192,4,144,244,1
72,192,2,185,32,19,157,61,3,138,162
830 DATA64,160,3,32,189,255,169,2,162,8,
168,32,186,255,32,192,255,174,192,2
840 DATA189,64,19,160,0,132,251,133,252,
162,2,32,198,255,32,165,255,32,9,14
850 DATA166,144,240,246,32,136,14,8,32,2
04,255,169,2,32,195,255,40,144,5,32
860 DATA209,14,176,44,174,192,2,56,165,2
51,233,1,157,128,19,165,252,233,0,157
870 DATA96,19,232,236,193,2,176,28,24,12
5,160,19,176,22,238,192,2,238,194,2
880 DATA165,252,24,105,1,157,64,19,169,1
5,32,195,255,76,28,12,96,173,192,2
890 DATA56,237,194,2,141,192,2,169,90,16
0,17,32,11,15,32,245,14,32,75,14,169
900 DATA121,160,17,32,11,15,172,192,2,19
0,0,19,173,192
910 REM
920 REM *** BLOCK 6 ***
930 REM
940 DATA2,32,233,13,234,177,65,234,32,21
0,255,200,202,208,246,32,28,15,174
950 DATA192,2,189,0,19,133,251,173,192,2
,32,233,13,162,0,177,65,157,64,3,232
960 DATA200,198,251,208,245,160,0,185,24
6,18,157,64,3,200,232,192,4,144,244
970 DATA172,192,2,185,32,19,157,61,3,138
,162,64,160,3,32,189,255,169,2,162
980 DATA8,168,32,186,255,32,192,255,174,
192,2,160,0,189,64,19,132,251,133,252
990 DATA189,128,19,133,253,189,96,19,133
,254,162,2,32,201,255,32,26,14,32,168
1000 DATA255,165,253,197,251,165,254,229
,252,176,240,32,204,255,169,2,32,195
1010 DATA255,32,136,14,144,5,32,209,14,1
76,16,44,195,2,48,3,76,213,13,206,194
```

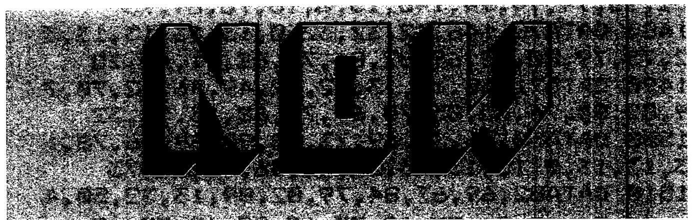



```
1020 DATA2,48,6,238,192,2,76,254,12,238,
192,2,174,192,2,236,193,2,176,4,32,240
1030 DATA14,96,169,133,160,17,32,11,15,3
2,245,14,104,104,76,16,8,32,117,11,206
1040 DATA194,2,16,3,76,183,13,238,192,2,
32,34,14,76,244,12,160,0,10,10,132,66
1050 DATA10,38,66,10,38,66,133,65,165,66
,24,105,20,133,66,160,0,96,120,160,52
1060 DATA132,1,160,0,96,32,1,14,145,251,
230,251
1070 REM
1080 REM *** BLOCK 7 ***
1090 REM
1100 DATA208,2,230,252,160,55,132,1,88,9
6,32,1,14,177,251,76,14,14,169,12,160
1110 DATA18,32,11,15,173,192,2,10,10,10,
10,8,174,192,2,188,0,19,170,189,0,20
1120 DATA40.8.144,3,189,0,21,32,210,255,
232,136,208,239,40,96,169,15,162,8,168
1130 DATA32,186,255,169,1,162,239,160,18
,32,189,255,32,192,255,169,15,76,195
1140 DATA255,32,165,255,41,15,10,10,10,1
0,133,87,32,165,255,41,15,5,87,96,72
1150 DATA74,74,74,74,32,128,14,104,41,15
,24,105,48,76,210,255,169,15,162,8,168
1160 DATA32,186,255,169,0,32,189,255,32,
192,255,32,163,14,8,169,15,32,195,255
1170 DATA40,96,162,15,32,198,255,32,100,
14,201,1,8,144,11,72,32,240,14,32,240
1180 DATA14,104,32,119,14,32,165,255,201
,13,240,10,40,8,144,245,32,210,255,24
1190 DATA144,239,32,204,255,40,96,169,22
3,160,17,32,11,15,32,245,14,201,49,208
1200 DATA2,24,96,201,50,208,243,56,96,32
,228,255,240,251,96,169,32,44,169,13
1210 DATA76,210,255,32,231,14,201,95,240
,1,96,169,15,32,195,255,32,231,255,162
1220 DATA247,154,76,16,8,133,87,132,88,1
60
1230 REM
1240 REM *** BLOCK 8 ***
1250 REM
1260 DATA0,177,87,240,6,32,210,255,200,2
08,246,96,162,0,142,64,3,142,65,3,174
1270 DATA192,2,189,160,19,162,8,72,104,1
0,72,248,173,65,3,109,65,3,141,65,3
1280 DATA173,64,3,109,64,3,141,64,3,216,
202,208,230,104,173,64,3,9,48,141,64
1290 DATA3,173,65,3,41,15,9,48,141,66,3,
173,65,3,74,74,74,74,9,48,141,65,3,169
1300 DATA29,133,211,162,0,189,64,3,201,4
8,208,10,169,32,157,64,3,232,224,3,144
1310 DATA239,162,0,189,64,3,32,210,255,2
32,224,3,144,245,96,147,13,13,32,32
1320 DATA32,32,32,32,32,42,42,42,42,32,8
3,85,80,69,82,32,67,79,80,89,32,49,53
1330 DATA52,49,32,42,42,42,42,13,32,32,3
2,32,32,32,32,40,67,41,32,66,89,32,78
1340 DATA46,77,65,78,78,32,38,32,68,46,8
7,69,73,78,69,67,75,13,13,13,13,32,32
1350 DATA49,46,32,68,73,82,69,67,84,79,8
2,89,13,32,32,50,46,32,75,79,80,73,69
1360 DATA82,69,78,13,32,32,51,46,32,70,7
9,82,77,65,84,73,69,82,69,78,13,32,32
1370 DATA52,46,32,83,67,82,65,84,67,72,1
3,32,32
1380 REM
1390 REM *** BLOCK 9 ***
1400 REM
1410 DATA53,46,32,86,65,76,73,68,73,69,8
2,69,78,13,32,32,54,46,32,69,78,68,69
1420 DATA13,13,13,13,32,32,18,32,66,73,8
4,84,69,32,87,65,69,72,76,69,78,32,83
1430 DATA73,69,32,58,32,146,32,0,13,13,3
2,70,73,76,69,32,90,85,32,76,65,78,71
1440 DATA0,13,13,32,18,32,75,79,80,73,69
,82,76,73,83,84,69,32,86,79,76,76,32
1450 DATA146,0,74,65,47,78,69,73,78,157,
157,157,157,157,157,157,0,18,32,74,65
1460 DATA32,146,32,32,32,0,13,32,32,18,3
2,70,65,76,83,67,72,69,82,32,70,73,76
1470 DATA69,84,89,80,32,146,32,32,32,32,
32,18,94,94,94,146,13,0,18,78,69,73
1480 DATA78,146,32,32,32,0,13,13,13,13,3
2,75,79,80,73,69,82,86,79,82,71,65,78
1490 DATA71,58,13,13,32,32,49,46,32,70,7
9,82,84,76,65,85,70,69,78,68,13,32,32
1500 DATA50,46,32,69,73,78,90,69,76,78,1
3,13,13,0,13,13,13,32,49,46,32,68,73
1510 DATA82,69,67,84,79,82,89,13,32,50,4
6,32,86,65,76,73,68,73,69,82,69,78,13
1520 DATA32,51,46,32,70,79
1530 REM
1540 REM *** BLOCK 10 ***
1550 REM
1560 DATA82,77,65,84,73,69,82,69,78,13,0
,13,13,32,86,65,76,73,68,73,69,82,69
1570 DATA78,32,46,46,46,13,0,13,13,32,18
,32,81,85,69,76,76,45,68,73,83,75,69
1580 DATA84,84,69,32,69,73,78,76,69,71,6
9,78,32,146,13,0,13,13,32,82,69,65,68
1590 DATA73,78,71,32,0,13,13,32,18,32,90
,73,69,76,45,68,73,83,75,69,84,84,69
1600 DATA32,69,73,78,76,69,71,69,78,32,1
46,13,0,13,13,32,87,82,73,84,73,78,71
1610 DATA32,0,13,13,32,32,18,32,75,32,79
,32,80,32,73,32,69,32,32,32,70,32,69
1620 DATA32,82,32,84,32,73,32,71,32,33,3
2,146,13,0,13,13,32,32,68,73,83,75,78
1630 DATA65,77,69,78,32,85,78,68,32,73,6
8,32,69,73,78,71,69,66,69,78,13,13,0
1640 DATA13,13,32,78,79,67,72,32,69,73,7
8,32,86,69,82,83,85,67,72,32,63,13,0
1650 DATA13,13,32,32,49,46,32,85,69,66,6
9,82,83,80,82,73,78,71,69,78,32,63,13
1660 DATA32,32,50,46,32,78,69,85,69,82,3
2,86,69,82,83,85,67,72,32,63,13,0,13
1670 DATA13,32,78
1680 REM
1690 REM *** BLOCK 11 ***
1700 REM
1710 DATA65,69,67,72,83,84,69,83,32,70,7
3,76,69,32,58,0,13,32,32,18,32,83,67
1720 DATA82,65,84,67,72,45,68,73,83,75,3
2,69,73,78,76,69,71,69,78,32,146,13
1730 DATA0,13,13,32,18,32,83,73,67,72,69
,82,32,63,32,45,32,83,80,65,67,69,44
1740 DATA32,83,79,78,83,84,32,95,32,146,
13,0,13,13,32,83,67,82,65,84,67,72,73
1750 DATA78,71,32,0,32,66,76,79,69,67,75
,69,32,90,85,32,75,79,80,73,69,82,69
1760 DATA78,32,146,13,0,32,66,76,79,69,6
7,75,69,32,90,85,32,83,67,82,65,84,67
1770 DATA72,69,78,32,146,13,0,13,32,42,4
2,42,32,83,80,65,67,69,32,42,42,42,13
1780 DATA32,32,70,85,69,82,32,87,69,73,8
4,69,82,13,13,32,18,32,66,73,84,84,69
1790 DATA32,87,65,69,72,76,69,78,32,83,7
3,69,32,146,13,0,32,52,46,32,78,79,67
1800 DATA72,77,65,76,32,75,79,80,73,69,8
2,69,78,13,0,36,0,73,0,44,88,44,82,0
1810 DATA44,88,44,87,0
READY.
```

»Super Copy« (Fortsetzung und Schluß)







Im Universum versteckt sich die interplanetarische Spinne Plexar. Sie können die Katastrophe abwenden, wenn Sie das All von dem Ungeheuer befreien.

Bevor Sie mit dem apokalyptischen Kampf gegen die interplanetarische Spinne beginnen, müssen Sie zuerst auftanken. Doch nach 30 Energieeinheiten beginnt ein feindlicher Hubschrauber auf Sie zu schießen. Ist dieses Hindernis hinter Ihnen, geht es links weiter. Im oberen Eck erscheint ein Tunnel, den Sie durchfliegen. Dabei müssen Sie eine Lasersperre überwinden oder einen Mann retten. Dieser sollte bei der Aktion möglichst weit links stehen. Haben Sie diese Hürde überwunden, schließt sich hinter Ihnen eine Mauer. Durch diese müssen Sie sich links unten durchschießen. Auf dem nächsten Bild gilt es eine Barrikade zu durchschießen, die aber mit Vorsicht zu genießen ist. Die nachfolgenden Panzer können Sie nur abschießen, wenn Sie beide zugleich treffen.

Dann kommen Sie zu der Spinne Plexar. Achten Sie vor allem auf die Lasersperren. Wenn Sie nun die Spinne vernichtet haben, fliegen Sie zurück, aber achten Sie darauf, daß Sie möglichst hoch in den nächsten Bildschirm fliegen.

Zum guten Schluß können Sie auf dem Energielager landen und sich von den Strapazen erholen oder Sie spielen weiter. Das hat aber nur Sinn, wenn Sie noch zwei Leben haben.

(Helmut Burgemeister/Helmut Bölcskei/rg)



Variablenliste »Apocalypse Now«

Die wichtigsten Variablen:

X und Y für Hubschrauber
X1 bis X7 und Y1 bis Y7 für andere Sprites
HU = feindliche Hubschrauberzahl
LD = Plexar abgeschossen?
A = Hubschrauberstellung
E = Schußvariable

Listing »Apocalypse Now« (Anfang)

```
1 REM +++++++++++++++++
2 REM + APOCALYPSE NOW +
3 REM +++++++++++++++++++++++++
4 REM * BY HELMUT BURGEMEISTER *
5 REM * AND HELMUT BOELCSKEI   *
6 REM *************************
7 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^
8 REM ^   MASCHINEN PROGRAMM    ^
9 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^
10 PRINT"▒▒▒▒▒▒BITTE WARTEN !"
20 POKE53281,0
30 POKE53280,0
40 FORI=832TO832+33:READA:POKEI,A:NEXT
50 DATA120,169,51,133,1,169,0,133,95,133
,90,133,88,169,208,133,96,169,240
60 DATA133,89,169,224,133,91,32,191,163,
169,55,133,1,88,96
80 SYS832:POKE850,160:SYS832
100 REM ______________________
101 REM _ DATEN FUER NEUE ZEICHEN _
102 REM ______________________
130 POKE56576,PEEK(56576)AND252
135 POKE53272,8:POKE648,192
140 PRINT"▒▒▒▒▒▒BITTE WARTEN !"
150 FORI=280TO559:READA:POKE57344+I,A:NE
XT
210 DATA255,255,255,255,255,255,255,255
220 DATA0,0,0,37,205,255,255,255
230 DATA0,0,0,0,0,255,255,255
240 DATA137,66,36,24,153,36,66,137
245 DATA255,129,129,255,129,129,255,255
250 DATA3,13,49,193,193,49,13,3
255 DATA0,96,252,255,252,96,0,0
260 DATA60,60,60,24,24,0,0,0
265 DATA255,126,255,126,255,126,255,126
270 DATA0,0,0,255,255,255,0,0
275 DATA24,60,24,63,88,88,36,66
280 DATA25,61,25,62,88,88,36,66
285 DATA0,0,60,32,60,32,32,32
290 DATA0,0,68,68,68,68,68,124
295 DATA0,0,124,64,124,64,64,124
300 DATA0,0,32,32,32,32,32,60
305 DATA126,255,255,231,231,255,255,126
310 DATA0,0,0,0,8,5,3,7
315 DATA0,0,0,0,12,62,127,255
320 DATA0,0,6,31,127,255,255,255
325 DATA0,0,0,192,231,239,255,255
330 DATA0,0,56,252,252,254,255,255
335 DATA255,127,127,63,31,12,0,0
340 DATA255,255,254,152,0,0,0,0
345 DATA255,191,31,15,3,1,0,0
350 DATA255,255,255,248,248,224,0,0
355 DATA0,0,36,24,24,36,0,0
360 DATA65,39,18,207,63,191,127,255
365 DATA160,68,88,225,250,252,254,255
```

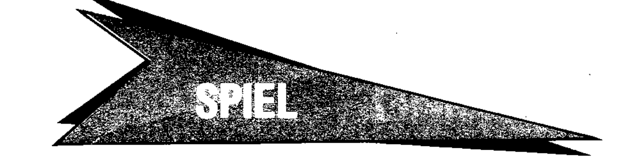
```
370 DATA255,255,255,255,255,255,255,255
375 DATA255,255,243,195,195,192,0,0
380 DATA0,0,35,254,35,0,0,0
385 DATA4,4,11,48,48,64,0,0
390 DATA0,0,196,127,196,0,0,0
395 DATA0,32,32,208,12,12,2,0
400 REM ++++++++++++++++
401 REM + SPRITES DATEN +
402 REM ++++++++++++++++
500 FORI=50304TO50304+62:READA:POKEI,A:N
EXT
505 FORI=50368TO50368+62:READA:POKEI,A:N
EXT
510 FORI=50432TO50432+62:READA:POKEI,A:N
EXT
511 FORI=50496TO50496+62:READA:POKEI,A:N
EXT
512 FORI=50560TO50560+62:READA:POKEI,A:N
EXT
513 FORI=50624TO50624+62:READA:POKEI,A:N
EXT
514 FORI=50688TO50688+62:READA:POKEI,A:N
EXT
515 FORI=50752TO50752+62:READA:POKEI,A:N
EXT
517 PRINT"▒   ▒ APOCALYPSE NOW "
518 PRINT"▒▒  ▒BY ▒HELMUT ▙BURGEMEISTER▒
AND HELMUT BOELCSKEI  "
519 DATA0,0,0
520 DATA0,0,0,255,255,255,0,48,0,3,255,1
93,4,32,97,8,32,55,16,42,153
525 DATA96,34,1,128,79,131,128,66,6,128,
138,152,255,128,96,64,64,128
530 DATA32,33,0,16,33,0,143,254,0,129,2,
0,127,255,248,0,0,0,0,0,0
540 DATA0,0,0,0,0,0,31,255,240,0,40,0,0,
16,0,0,104,0,0,132,0
545 DATA1,2,0,6,249,128,4,252,128,11,254
,64,9,252,64,4,0,128,3,3,0,0,252,0
550 DATA1,50,0,2,1,0,4,0,128,28,0,224,0,
0,0,0,0,0
560 DATA0,0,0,0,0,0,255,255,255,0,12,0,1
31,255,112,135,4,32
565 DATA252,4,16,153,68,8,128,68,6,193,2
42,1,96,66,1,24,65,1,6,1,255,1,2,2,0
570 DATA132,4,0,132,8,0,127,241,0,64,129
,31,255,254,0,0,0,0,0,0
580 DATA0,0,240,0,15,1,0,240,193,15,111,
103,240,240,65,3,32,145,4,42,3
585 DATA8,34,134,0,47,8,16,66,144,96,82,
96,128,128,128,128,129,0,143,65,0
590 DATA240,33,0,64,46,8,32,242,240,31,1
5,0,129,240,0,143,0,0,112,0,0
600 DATA15,0,0,128,240,0,131,15,0,230,25
2,240,156,240,15,137,4,192,192,84,32
605 DATA97,68,16,16,244,0,9,66,8,6,82,6,
1,1,1,0,129,1,0,130,241,0,132,15
610 DATA16,116,2,15,143,4,0,240,248,0,15
,129,0,0,241,0,0,14
620 DATA0,0,0,0,0,0,0,3,0,0,15,0,0,63,0,
0,240,0,0,192,0,0,20,0,0,20,0,0,40,0
625 DATA0,40,0,0,20,0,0,20,0,0,40,0,0,40
,0,0,20,0,0,95,0,0,170,0,3,255,192
630 DATA13,95,112,3,255,192
640 DATA0,0,0,0,0,0,192,0,0,240,0,0,63
,0,0,15,0,0,3,0,0,20,0,0,20,0,0,40,0
645 DATA0,40,0,0,20,0,0,20,0,0,40,0,0,40
,0,0,20,0,0,95,0,0,170,0,3,255,192
650 DATA13,95,112,3,255,192
660 DATA128,0,1,71,255,242,36,0,4,16,255
,12,25,0,148,38,0,100,2,0,68,69,36,196
665 DATA69,153,68,124,126,125,68,60,68,6
8,90,68,68,129,68,67,0,196,66,255,68,68
670 DATA20,36,40,34,20,16,65,24,47,255,2
28,64,0,2,128,0,1
700 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
701 REM ^ INETIALISIERUNG VARIABLEN ^
702 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
997 GOSUB62130:POKE50169,22:POKE50168,18
998 V=53248:POKEV+21,1:POKEV+42,14
999 X=250:Y=94:GOSUB50000
1000 X=250:Y=94:A=1:OT=1:V=53248:POKE501
68,18:GOSUB63000:O=PEEK(V+31):AB=AC
1001 POKEV+21,1:POKEV+29,3:POKEV+39,3:B=
1:PRINT"◙":Q=13:X2=0:Y2=90:B2=4:C2=3
1002 GOSUB60100:POKEV+21,1
1003 H=32:EN=10:PU=.05:B9=0:HU=0
1004 POKEV+2,X2:POKEV+3,Y2:HB=4
1005 POKEV+28,12:POKEV+37,2:POKEV+38,5
1006 POKE50170,23:POKE50171,24
1009 POKESI+11,65:JS=PEEK(V+30):O=PEEK(V
+31)
1010 REM HAUPTPROGRAMM
1011 J=PEEK(56320):JS=PEEK(V+30)
1020 IF(JAND4)=0THENX=X-5:A=A-1
1030 IF(JAND8)=0THENX=X+5:A=A+1
1040 IF(JAND1)=0THENY=Y-5
1050 IF(JAND2)=0THENY=Y+5
1055 IFA>3THENA=3
1056 IFA<1THENA=1
1059 IFY<85THENY=85
1060 ONAGOSUB2000,2010,2020
1065 Y=Y+AB
1070 POKEV,X:POKEV+1,Y
1080 IF(JAND16)=0THENGOSUB2050
1090 O=PEEK(V+31):IFO=1THEN3200
1098 IFX>250THENB=B-1:X=250:BI=0:HU=0:B9
=0:POKEV+21,1:IFB<1THENB=1
1099 IFX<5THENB=B+1:X=5:BI=0
1100 ONBGOSUB4500,5000,5500,6000
1110 POKE49152+EN+40*3,H:EN=EN-PU:H=32
1888 IFEN>10THEN1010
1889 GOSUB61000:GOSUB60900
1890 IFEN<=0THENEN=0:AB=4:POKESI+7,100:G
OTO1065
1900 GOTO1010
1997 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
1998 REM ^BEWEGUNGSRICHTUNG VON SPRITE^
1999 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
2000 IF(JAND4)=0THENPOKE50168,21:POKESI+
7,255:RETURN
2001 POKE50168,18:POKESI+7,200
2009 RETURN
2010 POKE50168,19
2019 RETURN
2020 IF(JAND8)=0THENPOKE50168,22:POKESI+
7,255:RETURN
2021 POKE50168,20:POKESI+7,200
2029 RETURN
2049 REM _________________
2050 REM _ SCHUSS ROUTINE _
```

Listing »Apocalypse Now« (Fortsetzung)

```
2051 REM __________________
2059 IF(JAND4)=0ANDA=1THENGOSUB3300:RETU
RN
2060 IFA=1THENGOSUB3000
2070 IFA=2THENGOSUB3030
2079 IF(JAND8)=0ANDA=3THENGOSUB3400:RETU
RN
2080 IFA=3THENGOSUB3060
2090 RETURN
3000 X1=INT(X/8-4):Y1=INT(Y/8-5)
3005 FORE=X1TO0STEP-1:GOSUB61400
3006 PO=PEEK(49152+E+40*Y1):POKE49152+E+
40*Y1,66:POKE49152+E+40*Y1,PO
3007 IFPEEK(49152+E-1+40*Y1)<>32THENS=-1
:GOTO3100
3010 NEXTE:RETURN
3030 X1=INT(X/8):Y1=INT(Y/8-3)
3033 FORE=Y1TO25STEP1:GOSUB61400
3040 PO=PEEK(49152+X1+40*E):POKE49152+X1
+40*E,42:POKE49152+X1+40*E,PO
3045 IFPEEK(49152+X1+40*E+40)<>32THEN311
0
3050 NEXTE:RETURN
3060 X1=INT(X/8+4):Y1=INT(Y/8-5)
3070 FORE=X1TO39STEP1:GOSUB61400
3075 PO=PEEK(49152+E+40*Y1):POKE49152+E+
40*Y1,68:POKE49152+E+40*Y1,PO
3077 IFPEEK(49152+E+1+40*Y1)<>32THENS=1:
GOTO3100
3080 NEXTE:RETURN
3097 REM ++++++++++++++++++++++
3098 REM + AUFPRALL VON SCHUSS +
3099 REM ++++++++++++++++++++++
3100 POKESI+10,3:GOSUB60700:IFPEEK(49152
+E+S+40*Y1)<>51THEN3150
3102 POKE49152+E+S+40*Y1,38:FORT=0TO100:
NEXT:POKE49152+E+S+40*Y1,32:E=0
3103 SC=SC+6:GOSUB61200:RETURN
3110 POKESI+10,3:GOSUB60700:IFPEEK(49152
+X1+40*E+40)<>51THEN3170
3111 POKE49152+X1+40*E+40,38
3112 FORT=0TO110:NEXT:POKE49152+X1+40*E+
40,32:E=0:SC=SC+8:GOSUB61200:RETURN
3150 PO=PEEK(49152+E+S+40*Y1):POKE49152+
E+S+40*Y1,38
3160 FORT=0TO100:NEXT:POKE49152+E+S+40*Y
1,PO:E=0
3165 RETURN
3170 PO=PEEK(49152+X1+40*E+40):POKE49152
+X1+40*E+40,38
3180 FORT=0TO100:NEXT:POKE49152+X1+40*E+
40,PO:E=0:RETURN
3197 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^
3198 REM ^ KOLLISION HUBSCHRAUBER ^
3199 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^
3200 GOSUB60200:POKESI+18,0:POKEV+2,0:X2
=0:Y2=INT(RND(1)*50)+90:HB=HB-1
3201 POKEV+3,Y2:O=PEEK(V+31):JS=PEEK(V+3
0)
3202 FORT=0TO100:P=INT(RND(1)*15):POKEV+
39,P:NEXT:POKEV+39,3
3203 IFHB<=0THENPOKEV+21,0:PRINT"
G A M E   O V E R":GOTO8000
3210 X=250:Y=94:POKEV,X:POKEV+1,Y:O=PEEK
(V+31):GOSUB60300:GOTO1009
3300 X1=INT(X/8-4):Y1=INT(Y/8-5)
```

Listing »Apocalypse Now« (Fortsetzung)

```
3310 FORE=X1TO0STEP-1:GOSUB61400
3315 Y1=Y1+.5
3316 IFY1>=24THENY1=24:RETURN
3319 PO=PEEK(49152+E+40*INT(Y1))
3320 POKE49152+E+40*INT(Y1),67
3321 POKE49152+E+40*INT(Y1),PO
3330 IFPEEK(49152+E-1+40*INT(Y1))<>32THE
NS=-1:GOTO3100
3340 NEXTE:RETURN
3400 X1=INT(X/8+4):Y1=INT(Y/8-5)
3410 FORE=X1TO39STEP1:GOSUB61400
3415 Y1=Y1+.5
3416 IFY1>=24THENY1=24:RETURN
3419 PO=PEEK(49152+E+40*INT(Y1))
3420 POKE49152+E+40*INT(Y1),69
3421 POKE49152+E+40*INT(Y1),PO
3430 IFPEEK(49152+E+1+40*INT(Y1))<>32THE
NS=1:GOTO3100
3440 NEXTE:RETURN
3997 REM _________________________
3998 REM _ BILD TANKEN AUFBAUEN _
3999 REM _________________________
4000 GOSUB60500:X2=0:Y2=90:POKEV+2,X2:PO
KEV+3,Y2
4001 X=250:Y=94:BI=1
4002 TR=49152:C=40:GOSUB4530
4009 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
4010 REM ^ BILDSCHIRMABFRAGE TANKEN ^
4011 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
4500 IFBI=0THEN4000
4501 IFB9=0THEN:IFEN>30ANDHU<5THENB9=1:P
OKEV+21,3:GOSUB61570:X3=20
4502 ONB9GOSUB61300
4503 IFY<=175ANDKL=0THENRETURN
4504 IFX>140ANDX<175ANDKL=0THENKL=1:GOSU
B4511:IFLD=1THENGOTO62000
4505 PU=.05:IFY>214ANDKL=1THENH=160:OT=1
:AB=AC:PU=-1:GOSUB60800
4506 IFY>218ANDKL=1THENY=Y-AB-1
4507 IFY<=170ANDKL=1THENKL=0:GOSUB4530:R
ETURN
4508 IFY>214ANDKL=1ANDEN>=35THENEN=35:H=
32:POKESI+18,0
4509 IFY<=214ANDKL=1THENPOKESI+18,0
4510 RETURN
4511 FORT=18TO23
4512 FORI=15TO24
4515 POKETR+I+40*T,32
4520 NEXT:NEXT
4521 PRINT""
4522 PRINT"            >#####  ________
####?
4523 PRINT"****************#            #
#############
4529 RETURN
4530 PRINT""
4531 PRINT"                    %%%%%%%%%%
4532 PRINT"                  >###/012###?
4533 PRINT"                >##''''''''''
#?
4534 PRINT"               >###''''''''''
##?
4536 PRINT"              >####''''''''''
###?
4537 PRINT"             >#####''''''''''
####?
```




```
4538 PRINT"▒$$$$$$$$$$$$#####▒'''''''''''▒#
#####$$$$$$$$$▒▒
4560 KL=0:RETURN
4997 REM ________________________
4998 REM _ BILD TUNNEL AUFBAUEN _
4999 REM ________________________
5000 IFBI=0THENH=32:GOTO5010
5005 GOTO5300
5010 GOSUB60500:TI$="000000"
5011 X=250:Y=94:BI=1:PX=45:PY=46
5019 PRINT"▒▒▒▒▒"
5020 PRINT"     >6677656756567756765667 87
8776767657
5030 FORT=0TO3:PRINT"     ()":NEXT
5035 PRINT"     ()           678687688767876
8676787867
5040 FORT=0TO2:PRINT"▒      ()
     ▒++":NEXT
5045 PRINT"▒ (67867887876767676668787
     ▒++
5050 PRINT"▒ (9:▒)))))))))))))))))▒9;9:
     ▒++"
5085 PRINT"▒ (            ▒333         3
     ▒++
5090 PRINT"▒ (             ▒3          3
     ▒++
5095 PRINT"▒ (                         ▒3
     ▒++
5096 PRINT"▒ (            ▒33333       ▒3
     ▒++
5100 PRINT"▒ 788▒3333333▒88878787878787
     ▒++
5110 PRINT"▒ ;:9▒3333333▒;:9:9:;9:9:;9:
     ▒++
5120 PRINT"                   ▒   /\
     ▒++
5125 PRINT"                   ▒  /  \
     ▒++
5130 PRINT"                   ▒ /    \
     ▒++
5135 PRINT"▒$$$$$$$$$$$$▒#▒$$$$$$▒#▒$$$$
$$$$$$$$$▒++▒▒
5297 REM ++++++++++++++++++++++++++++
5298 REM + BILDSCHIRMABFRAGE TUNNEL +
5299 REM ++++++++++++++++++++++++++++
5300 POKETR+Q+C*17,PX
5320 POKETR+Q+C*17,PY:IFY<170THENRETURN
5340 GE=INT(RND(1)*15):IFGE>10THENGOSUB6
0000
5345 IFX<140ANDX>120ANDY>174ANDY<178THEN
PX=32:PY=32:GOSUB61100:GOSUB61200
5350 IFTI$>"000005"THENGOSUB60070
5407 IFY<=214THENPOKESI+18,0
5410 RETURN
5497 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
5498 REM ^ BILDSCHIRM PANZER AUFBAUEN ^
5499 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
5500 IFBI=0THENH=32:GOTO5510
5505 GOTO5700
5510 GOSUB60500
5520 X=250:Y=94:BI=1:B5=-2:POKEV+5,223:P
OKEV+7,223:X5=50:X6=120:POKEV+21,13
5521 POKEV+4,X5:POKEV+6,X6:GOSUB5800
5530 PRINT"▒▒▒▒▒":B6=2:Y7=20:G9=1
5540 PRINT"767878787878786868887868786 87
8687868786"
5545 PRINT":;:;:;9:;9;9:9:;9:;9
     "
5547 PRINT"+)                  (+"
5550 FORT=0TO2:PRINT"+)                 (
+▒333333▒":NEXT
5555 PRINT"+)          +        (+▒33333333
3▒7786876787"
5560 FORT=0TO3:PRINT"▒ 3▒           +
 (+▒333333333▒+)":NEXT
5565 FORT=0TO5:PRINT"▒+)            ▒+
 ▒333   333333▒+)":NEXT
5570 FORT=0TO2:PRINT"▒
(+▒333333333▒+)":NEXT
5575 PRINT"▒$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$▒▒"
5697 REM ++++++++++++++++++++++++++++
5698 REM + BILDSCHIRMABFRAGE PANZER +
5699 REM ++++++++++++++++++++++++++++
5700 IFX>168THENRETURN
5710 IFX5>138THENB5=3:POKE50170,24
5715 IFX5<30THENB5=-2:POKE50170,23
5720 IFX6>138THENB6=2:POKE50171,24
5730 IFX6<30THENB6=-3:POKE50171,23
5740 X5=X5-B5:X6=X6-B6
5750 POKEV+4,X5:POKEV+6,X6
5755 JS=PEEK(V+30):IFJS=5ORJS=9ORJS=13TH
EN3200
5765 ONG9GOSUB5820,5900
5770 RETURN
5800 X7=INT(X5/8-1):Y7=21
5810 RETURN
5811 X8=INT(X6/8-1):Y8=21
5815 RETURN
5820 Y7=Y7-1
5835 PO=PEEK(49152+X7+40*Y7-40)
5840 POKE49152+X7+40*Y7,90
5850 IFPO<>32THENPOKE49152+X7+40*Y7-40,P
O:POKE49152+X7+40*Y7,32:G9=2:GOSUB5811
5860 POKE49152+X7+40*Y7,32
5870 RETURN
5900 Y8=Y8-1
5920 PO=PEEK(49152+X8+40*Y8-40)
5930 POKE49152+X8+40*Y8,90
5940 IFPO<>32THENPOKE49152+X8+40*Y8-40,P
O:POKE49152+X8+40*Y8,32:G9=1:GOSUB5800
5950 POKE49152+X8+40*Y8,32
5960 RETURN
5997 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
5998 REM ^ BILDSCHIRM SPINNE AUFBAUEN ^
5999 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
6000 IFBI=0THENH=32:GOTO6020
6010 GOTO6300
6020 GOSUB60500:TI$="000000":V=53248:POK
E50172,25:POKEV+8,50:POKEV+9,157
6030 X=250:Y=110:BI=1:POKEV+23,16:POKEV+
29,19:POKEV+21,17
6040 POKEV,X:POKEV+1,Y:O=PEEK(V+31)
6050 PRINT"▒▒▒▒▒"
6060 PRINT"                 ▒67878786786878 68
76878677867
6070 PRINT"                 ▒+;9:;9:9;9;9:9;+
9:;:;♣;9:;9
6080 FORT=0TO1:PRINT"              ▒+)
     (+":NEXT
6090 FORT=0TO2:PRINT"              +)
     ":NEXT
```

Listing »Apocalypse Now« (Fortsetzung)

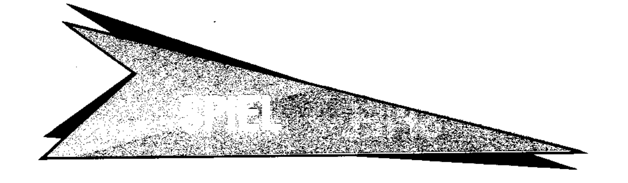
```
6100 PRINT" 76768787868+
6110 PRINT"(+++++++++++                    +
) ^ACHTUNG^
6120 PRINT"(++          ++)                 (+)
6130 PRINT"(++          ++                   +)
6140 PRINT"(+            333
+)
6150 PRINT"(+            333
+)
6160 PRINT"(++          ++                   +)
6170 PRINT"(++          ++)                 (+)
6180 PRINT"(+++++++++++7                    +
)
6190 PRINT" ::;9:;:9;9:+
+)
6200 PRINT"                 +)      +(       +
)
6210 PRINT"              +)      +(
+)
6220 PRINT"              +)      +(     /012
+)
6230 PRINT"$$$$$$$$$$$$$+$$$$$$+$$$$$
$$$+$$$$$$$$$$$"
6297 REM ++++++++++++++++++++++++++++
6298 REM + BILDSCHIRMABFRAGE SPINNE +
6299 REM ++++++++++++++++++++++++++++
6300 IFTI$>"000007"THENGOSUB6400
6305 IFTI$>"000010"THENGOSUB6420
6310 PU=.05:IFY>206ANDX>180ANDX<190THENG
OSUB6440
6320 IFY<=206THENPOKESI+18,0
6325 O=PEEK(V+31):IFO=16THENGOSUB6550
6399 RETURN
6400 PRINT"
"
6405 O=PEEK(V+31):IFO=1THEN3200
6410 PRINT"
"
6415 RETURN
6420 PRINT"
"
6425 O=PEEK(V+31):IFO=1THEN3200
6430 PRINT"
"
6435 TI$="000000":RETURN
6440 IFHA=1THENRETURN
6441 PU=.05:OT=1:H=160:PU=-1:GOSUB60800:
IFEN>35THENEN=35:HA=1:H=32:PU=.05:OT=1
6450 IFY>209THENY=Y-AB-1
6460 RETURN
6550 FORT=0TO180:POKE53280,T:POKE53281,T
:GOSUB60700:NEXT:POKE53280,0
6552 POKE53281,0:LD=1:SC=SC+1000:GOSUB61
200
6560 POKEV+21,1:RETURN
6697 REM ++++++++++++++++++
6698 REM + PUNKTE WERTUNG +
6699 REM ++++++++++++++++++
8000 FORT=0TO1500:NEXT:PRINT"
NOCH EINMAL [J] "
8002 PRINT"SCORE "
;SC
8003 IFSC>ASTHENAS=SC
8004 PRINT"HIGHSCOR
E ";AS
8010 POKE198,0:WAIT198,1:GETA$:IFA$="J"T
HEN9000
8015 POKE648,4:SYS65126
8020 PRINT" DANKE!!! GOOD BYE !!!!!!!!!
":END
9000 BI=0:B9=0:EN=0:HU=0:PU=0:AB=0:KL=0:
H=0:PX=0:PY=0:GE=0:PO=0:G9=0
9010 B=0:HA=0:LD=0:SC=0:AC=0:POKEV+23,0:
POKEV+29,0:GOTO998
10000 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^
10001 REM ^ ANFANGSSPRITE BEWEGEN ^
10002 REM ^^^^^^^^^^^^^^^^^^^^^^^^^^
50000 POKEV,X:POKEV+1,Y
50010 X1=INT(X/8-4):Y1=INT(Y/8-5)
50020 FORE=X1TO0STEP-1
50030 POKE49152+E+40*Y1,66:POKE49152+E+4
0*Y1,32
50040 IFPEEK(49152+E-1+40*Y1)<>32THENPOK
E49152+E-1+40*Y1,38:GOTO50070
50050 NEXTE:RETURN
50070 FORT=0TO100:NEXT:POKE49152+E-1+40*
Y1,32:X=X-3:E=0:GOTO50000
60000 POKETR+Q+C*17,32:Q=Q-INT(RND(1)*3)
:Q=Q+INT(RND(1)*3)
60010 IFQ<11THENQ=11
60020 IFQ>14THENQ=14
60060 RETURN
60070 OT=OT*-1:TI$="000000"
60075 PRINT""
60080 IFOT=-1THENFORT=0TO2:PRINT"
":NEXT:RETURN
60085 PRINT"                /\    "
60086 PRINT"               /  \    "
60087 PRINT"              /    \ "
60090 RETURN
60099 REM GERAEUSCHE
60100 SI=54272:POKESI+7,200:POKESI+8,0:P
OKESI+9,0:POKESI+10,3:POKESI+11,0
60110 POKESI+12,0:POKESI+13,15*16+15:POK
ESI+21,0:POKESI+22,150:POKESI+23,243
60120 POKESI+24,31:POKESI+11,65
60130 RETURN
60200 POKESI+11,0
60201 POKESI+1,5:POKESI+4,0:POKESI+5,11:
POKESI+6,0
60202 POKESI+4,131
60210 RETURN
60300 IFEN<=1THENEN=15:IFB>1THENEN=35
60301 PRINT"
"
60302 PRINT"L
EBEN "HB
60303 PRINT"ENERGY":FORT=0TOEN:PO
KE49152+T+40*3,160:NEXT
60304 PRINT"                   "
60310 X=250:Y=94:AB=AC:H=32:GOSUB61200
60315 IFB=4THENY=110
60320 RETURN
60500 PRINT"
"
60501 PRINT"ENERGY"
60502 FORT=0TOEN:POKE49152+T+40*3,160:NE
XT:GOSUB61200
60550 RETURN
60700 SI=54272:POKESI+4,0:POKESI+1,10:PO
KESI+5,12:POKESI+6,0:POKESI+4,131
60710 RETURN
```

Listing »Apocalypse Now« (Fortsetzung)




```
60720 SI=54272:POKESI+4,0:POKESI+1,20:PO
KESI+5,12:POKESI+6,0:POKESI+4,131
60730 RETURN
60800 SI=54272:POKESI+18,0:POKESI+19,0:P
OKESI+20,15*16
60810 POKESI+14,200:POKESI+15,39:POKESI+
18,17
60820 RETURN
60900 SI=54272:POKESI+18,0:POKESI+19,0:P
OKESI+20,15*16+2
60910 POKESI+14,200:POKESI+15,59:POKESI+
18,17
60920 RETURN
61000 OT=OT*-1:IFOT=-1THENPRINT"
           ":RETURN
61001 PRINT"LOW ON FUEL
"
61002 RETURN
61100 SC=SC+100:PRINT"
++++"
61120 RETURN
61200 PRINT"
SCORE ";SC;""
61210 RETURN
61297 REM ++++++++++++++++++++++++++++
61298 REM + FEINDHUBSCHRAUBER BEWEGEN +
61299 REM ++++++++++++++++++++++++++++
61300 IFX>X2THENB2=+5:M4=22
61310 IFX<X2THENB2=-5:M4=21
61320 IFY>Y2THENC2=+3
61330 IFY<Y2THENC2=-3
61335 X2=X2+B2:Y2=Y2+C2
61340 POKEV+2,X2:POKEV+3,Y2
61341 O=PEEK(V+31)
61342 JS=PEEK(V+30)
61350 IFJS=3THEN3200
61351 IFO=2THEN61500
61352 JS=PEEK(V+30)
61353 IFHU>3THENGOSUB61530
61354 O=PEEK(V+31)
61355 POKE50169,M4
61360 RETURN
61400 IFB=3THEN61430
61401 IFB=4THEN61470
61402 IFHU>4THENRETURN
61410 O=PEEK(V+31):IFO=2THEN61500
61420 RETURN
61430 O=PEEK(V+31):IFO=12THENGOSUB61450:
POKEV+21,1:SC=SC+200:GOSUB61200:G9=0
61440 RETURN
61450 GOSUB60720:FORT=0TO100:POKEV+38,T:
NEXT:RETURN
61470 O=PEEK(V+31):IFO=16THEN:GOSUB6550
61480 RETURN
61500 HU=HU+1:SC=SC+100:GOSUB61200
61501 GOSUB60700
61502 FORT=0TO100:POKEV+40,T:NEXT:X2=0:P
OKEV+2,X2:Y2=INT(RND(1)*50)+90
61503 POKEV+3,Y2
61504 IFHU>4THENPOKEV+21,1:B9=0
61510 E=0:RETURN
61530 X3=X3+1
61535 IFX3>X/8THENGOSUB61570
61536 IFX3>39THENGOSUB61570
61540 POKE49152+X3+40*INT(Y3),68
61545 O=PEEK(V+31):IFO=1THENPOKE49152+X3
+40*INT(Y3),32:GOTO3200
61550 POKE49152+X3+40*INT(Y3),32
61560 RETURN
61570 X3=INT(X2/8+4):Y3=INT(Y2/8-5)
61580 RETURN
61997 REM _________
61998 REM _ MUSIK _
61999 REM _________
62000 FORI=54272TO54296:POKEI,0:NEXT:POK
E54296,15:POKE54275,3
62110 H1=54272:H2=H1+1:AD=54277:SR=AD+1:
DR=AD-1:W1=64:W2=32
62120 POKEAD,72:POKESR,122:POKEAD+7,43:P
OKESR+7,187:GOTO62250
62130 DIMF$(18),FL(18),FH(18):FORI=0TO18
62140 READF$(I),F:FH(I)=INT(F/256):FL(I)
=F-FH(I)*256:NEXT
62150 DIMN1$(200),D1(200),N2$(200),D2(20
0):I=0
62160 READN1$(I),D1(I):IFD1(I)>0THENI=I+
1:GOTO62160
62170 I=0
62180 READN2$(I),D2(I):IFD2(I)>0THENI=I+
1:GOTO62180
62200 DIML1(200),H1(200),L2(200),H2(200)
:I=0
62210 FORJ=0TO18
62220 IFF$(J)=N1$(I)THENL1(I)=FL(J):H1(I
)=FH(J)
62230 IFF$(J)=N2$(I)THENL2(I)=FL(J):H2(I
)=FH(J)
62240 NEXT:I=I+1:IFN1$(I)<>""THEN62210
62245 RETURN
62250 SP=VAL("2")
62260 IFSP<1ORSP>4THEN62250
62270 S1=0:S2=0:C1=0:C2=0:POKEH1,L1(S1):
POKEH2,H1(S1)
62280 POKEH1+7,L2(S2):POKEH2+7,H2(S2):PO
KEDR,W1OR1:POKEDR+7,W2OR1
62290 C1=C1+1:IFC1>(D1(S1)*SP/2)THENPOKE
DR,W1
62300 IFC1<(D1(S1)*SP)THEN62320
62310 C1=0:S1=S1+1:POKEDR,W1OR1:POKEH1,L
1(S1):POKEH2,H1(S1)
62320 C2=C2+1:IFC2>(D2(S2)*SP/2)THENPOKE
DR+7,W2
62330 IFC2<(D2(S2)*SP)THEN62350
62340 C2=0:S2=S2+1:POKEDR+7,W2OR1:POKEH1
+7,L2(S2):POKEH2+7,H2(S2)
62350 IFD1(S1)>0THEN62290
62360 POKEV+21,0:GOTO8000
62370 DATAD3,2408,E3,2703,F#3,3034,G3,32
15
62380 DATAA3,3608,H3,4050,C#4,4547,D4,48
17
62390 DATAE4,5407,F#4,6069,G4,6403,A4,72
17
62400 DATAH4,8101,C#5,9094,D5,9634,E5,10
814
62410 DATAF#5,12139,G5,12860,A5,14435
62420 REM  1.STIMME
62430 DATAD5,3,D5,1,D5,1,E5,1,D5,1,C#5,1
,H4,4,H4,4,C#5,1,C#5,1
62440 DATAD5,1,E5,3,D5,1,E5,1,F#5,2,F#5,
2,E5,1,A4,3
```

Listing »Apocalypse Now« (Fortsetzung)




```
62450 DATAD5,3,D5,1,D5,1,E5,1,D5,1,C#5,1
,H4,4,H4,4,C#5,1,C#5,1
62460 DATAD5,1,E5,3,D5,1,E5,1,F#5,2,F#5,
2,E5,1,A4,3
62470 DATAF#5,2,A5,2,G5,2,F#5,1,D5,1,E5,
1,F#5,2,E5,5
62480 DATAF#5,3,E5,1,D5,2,D5,1,H4,1,E5,1
,F#5,2,E5,3,D5,1,E5,1
62490 DATAF#5,2,A5,2,G5,2,F#5,1,D5,1,E5,
1,C#5,1,H4,1,A4,5
62500 DATAD5,3,C#5,1,H4,1,C#5,1,D5,2,E5,
10
62510 DATAF#5,2,E5,6,F#5,2,E5,2,D5,1,E5,
1,F#5,2,A5,2,G5,2,F#5,1,D5,3,F#5,2,E5,6
62520 DATAF#5,2,E5,6,F#5,2,E5,2,D5,1,E5,
1,F#5,2,A5,2,G5,2,F#5,1,D5,1
62530 DATAE5,1,C#5,1,H4,1,A4,3,H4,1,C#5,
1
62540 DATAD5,3,D5,1,D5,1,E5,1,D5,1,C#5,1
,H4,4,H4,4,C#5,1,C#5,1
62550 DATAD5,1,E5,3,D5,1,E5,1,F#5,2,F#5,
2,E5,1,A4,3
62560 DATAD5,3,D5,1,D5,1,E5,1,D5,1,C#5,1
,H4,4,H4,4,C#5,1,C#5,1
62570 DATAD5,1,E5,3,D5,1,E5,1,F#5,2,F#5,
2,E5,1,D5,7
62580 DATA,-1
62590 REM  2.STIMME
62600 DATAD3,4,D3,4,G3,4,H3,2,D3,2,A3,4,
A3,4,D3,4,A3,2,E4,1,A3,1
62610 DATAD3,4,D3,4,G3,4,H3,2,D3,2,A3,4,
A3,4,D3,4,A3,2,E4,1,A3,1
62620 DATAD3,4,D3,4,A3,4,A3,4,H3,4,G3,4,
A3,4,A3,4
62630 DATAD3,4,D3,4,A3,4,A3,4,H3,4,G3,4,
A3,4,A3,3
62640 DATAA3,1,A3,7,A3,1,A3,8,D3,4,D3,4,
A3,4,A3,3
62650 DATAA3,1,A3,7,A3,1,A3,8,D3,4,D3,4,
A3,4,A3,4
62660 DATAD3,4,D3,4,G3,4,H3,2,D3,2,A3,4,
A3,4,D3,4,A3,2,E4,1,A3,1
62670 DATAD3,4,D3,4,G3,4,H3,2,D3,2,A3,4,
A3,4,D3,4,D3,8
62680 DATA,-1
62997 REM +++++++++++++++
62998 REM + GRAVITATION +
62999 REM +++++++++++++++
63000 PRINTCHR$(142):POKEV+21,0
63001 PRINT"[illegible]GRAVITAT
ION  [ 1 - 3 ]"
63010 POKE198,0:WAIT198,1:GETA$:IFA$="1"
THENAC=.1
63020 IFA$="2"THENAC=.5
63030 IFA$="3"THENAC=1
63040 IFA$<"1"ORA$>"3"THEN63010
63050 RETURN
READY.
```

Listing »Apocalypse Now« (Schluß)

Epedemic

Man schreibt das Jahr 1989. Ein Meteoritenschwarm droht mit der Erde zu kollidieren und eine weltweite Katastrophe auszulösen. Auf den Meteoriten befinden sich Viren, die eine tödliche Seuche verursachen. Es liegt an Ihnen, das Spiel mit möglichst wenig Verlusten an Menschenleben und einer unverseuchten Erde zu beenden.

Nach dem Laden und Starten des Programms erscheint zunächst die Frage, wieviele Verluste Sie sich zugestehen. Diese Zahl darf maximal vierstellig eingegeben werden. Danach erscheint die erste Bildschirmanzeige mit der Angabe des Tages (jede Spielrunde entspricht einem Spieltag), der Anzahl der anfliegenden Meteoriten und deren augenblickliche Entfernung von der Erde. Außerdem wird die Gesamtbevölkerung sowie die Verlustquote angezeigt.

Durch Betätigen der Funktionstaste F1 wechseln wir ins zweite Bild. Hier erhält man einen Überblick über den Stand der Dinge in den acht Regionen des Aktionsfeldes (Tabelle 1). Angezeigt wird für jede Region die Bevölkerungszahl sowie etwaige Verluste.

Durch nochmaliges Drücken von F1 gelangt man nun endlich ins Aktionsbild. Hier gibt es vier Aktionsmöglichkeiten, von denen jedoch nur eine pro Spieltag erlaubt ist.

Haben Sie sich für die medizinische Versorgung einer Region entschieden und den entsprechenden Menüpunkt ausgewählt, dann erscheint auf dem Bildschirm eine Übersicht über die verseuchten Regionen mit Schlüsselnummer und Level. Der Level ist die prozentuale Verlustquote der Bevölkerung. Dazu wird der Medikamentenvorrat angezeigt, und es erscheint die Frage nach der Region. Geben Sie an dieser Stelle die Schlüsselziffer der gewählten Region ein.

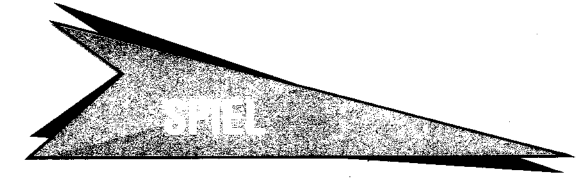
Es erscheint eine Angabe über die benötigte Menge an Medikamenten für die Versorgung dieser Region. Diese Menge wird von dem Gesamtvorrat an Medikamenten abgezogen und Sie kehren automatisch ins Hauptmenü zurück.

Sie können versuchen, einige der anfliegenden Meteore präventiv mit Raketen abzuschießen. Haben Sie sich hierfür entschieden, dann werden zunächst alle in Frage kommenden Meteore aufgelistet.

Achtung ! Der Wirkungsradius der Raketen beträgt zwischen 15000 und 50000 Kilometer. Meteore außerhalb dieses Entfernungsbereiches können nicht mit Raketen bekämpft werden.

Sie brauchen sich jetzt nur zu entscheiden, welchen Meteor Sie angreifen wollen. Der Raketenstart erfolgt vollautomatisch; allerdings trifft nicht jede Rakete ihr Ziel.

Falls eine der Regionen so hochgradig verseucht ist, daß selbst massiver Medikamenteneinsatz keine Aussicht auf Erfolg mehr verspricht, bleibt als letzte Lösung nur noch die nukleare Vernichtung der gesamten Region, um ein Übergreifen der Seuche auf andere Regionen zu verhindern. Wenn Sie sich dafür entschieden haben, erhalten Sie zunächst eine Übersicht über alle verseuchten Regionen, deren Bevölkerungszahl schon auf weniger als die Hälfte der ursprünglichen Anzahl gesunken ist.



Geben Sie jetzt die Schlüsselzahl der entsprechenden Region ein. Wenn Sie es sich in der Zwischenzeit doch anders überlegt haben, tippen Sie an dieser Stelle nur eine Null ein.

Die Produktion von Medikamenten ist die letzte der vier Aktionsmöglichkeiten. Auf dem Bildschirm erscheint eine Auflistung der unverseuchten Regionen und die dort jeweils produzierte Menge an Medikamenten. Diese Menge richtet sich nach der jeweiligen Bevölkerungsanzahl. Die Region mit der größten Bevölkerung hat auch den größten Ausstoß an Medikamenten. Beachten Sie, daß nur in völlig unverseuchten Regionen produziert wird.

Als reines Informationsbild gibt es die Weltgrafik. Auf dem Bildschirm wird eine Karte dargestellt, in die der Grad der Verseuchung in den einzelnen Regionen eingetragen ist. Die Verseuchung wird dabei in fünf Stufen angezeigt, nämlich unverseucht, bis 25, 50, 75 oder 100 Prozent verseucht. Da diese Weltgrafik nur zur Übersicht dient und Sie hier keine Entscheidung treffen können, gelangen Sie nach Betätigen einer Taste wieder zurück ins Menü, von wo aus Sie wieder alle vier Entscheidungsmöglichkeiten haben.

Die Meteore haben eine Geschwindigkeit von ungefähr 9000 Kilometern pro Tag. Hat man die erste Meteorwelle erfolgreich bekämpft, folgen weitere Meteore nach. Erst nach 30 Tagen Aktionszeit stürzen keine weiteren Meteore mehr auf die Erde ein, und das Spiel kann ausklingen. Nach 60 Tagen schließlich ist das Spiel auf jeden Fall beendet.

Ein Meteoraufschlag fordert zwei Millionen Todesopfer direkt und durch Verseuchung sofort weitere fünf Millionen Tote. Jeder weitere Tag, an dem die Seuche in der betreffenden Region nicht bekämpft wird, bedingt weitere fünf Millionen Tote.

Ist eine Region total verseucht, so können die Nachbarländer infiziert werden. Eine weitsichtige, strategische Planung der Seuchenbekämpfung ist daher unumgänglich.

Der Schwierigkeitsgrad des Spieles ist zugegebenermaßen nicht gerade niedrig. Nur wer die Nerven behält und weitsichtig plant, wird letzten Endes die völlige Auslöschung der Menschheit, zumindest im Spiel, verhindern können.

(Wolf-D. Robrahn/ev)

Region	Millionen Einwohner
Asien	2738
UdSSR	312
Europa	576
Afrika	480
Australien	24
Nordamerika	310
Südamerika	362
Wasser	—
gesamt	4802

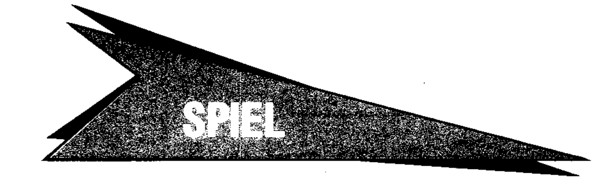
Tabelle 1. Das Aktionsfeld ist in acht Regionen aufgeteilt

```
10 rem"epidemic"
15 poke36879,168
20 printchr$(14)
25 print"██████████      1983 by"
30 print"█████Wolf-D.Robrahn"
35 print"█████Beethovenstr.18"
40 print"█████2200  Elmshorn"
50 fort=1to5000:next
51 printchr$(142)
52 print"████████████████████████████
█"
53 print"    e p e d e m i c"
54 print" ████████████████████"
55 poke36879,11
60 fora1=25to249step32
65 poke36879,a1
70 fort=1to1000:next
75 next
80 poke36879,25
100 forr1=1to200
110 r2=int(rnd(1)*126)+4250
120 poker2,81:poker2+33792,r3
130 r3=r3+1:ifr3=8thenr3=0
140 ifr3=1thenr3=2
150 next
155 print"█"
160 fora1=1to462
170 print"██ ";:next
190 printchr$(14)
200 print"██wieviele Verluste     geste
hen Sie sich bei Ihrer strategischen
Fuehrung";
210 print" ein?"
220 print"██Gesamtbev. "
230 print"█4802█ Mill.Einwohner"
240 print"█████████Mill.Einwohner"
250 input"█";a$
260 by=val(a$)
498 mz=50:rk=35
499 poke36879,25:printchr$(144)
500 rem anzahl meteor
520 a=13
530 dimme(a)
540 fora1=1toa
545 a3=int(rnd(1)*4e4)+9e4
550 me(a1)=a3
560 next
570 goto700
600 rem meteor entf.
620 fora1=1toa
630 a3=int(rnd(1)*8e4)+3e4
640 me(a1)=a3
650 next
655 ifbb=1then1400
700 rem variablen
730 da=1:po=4802:px=po
750 a$(5)=" AUSTRAL.:"
751 a$(1)="    ASIEN:"
752 a$(3)="   EUROPA:"
753 a$(6)="NORDAMER.:"
754 a$(7)="SUEDAMER.:"
755 a$(4)="   AFRIKA:"
756 a$(2)="    UdSSR:"
757 a$(8)="   WASSER:"
758 a$(9)="   WASSER:"
770 po(1)=2738:po(3)=576:po(6)=310:po(7
)=362:po(4)=480:po(2)=312:po(5)=24
775 fora1=1to7:px(a1)=po(a1):next:bb=1
815 printchr$(14)
950 fort1=1to3000:next
1400 rem total status
1415 fora1=1to9:ifp1(a1)=0thennext:goto
1500
1420 print"██Zustand fuer Tag:"da
1430 print"┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴"
1435 print"█ Region      Aufschl. "
1440 fora1=1to9
1445 ifp1(a1)=0then1460
1450 printa$(a1),"    ##"
1460 next
1465 print"███    weiter mit f1"
```

Listing »Epedemic«
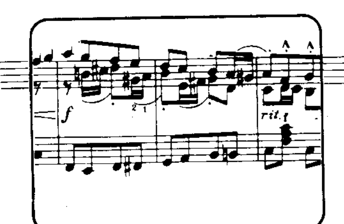
```
1470 getx$:ifx$<>chr$(133)then1470
1480 fora1=1to9:p1(a1)=0:next
1500 rem total status
1515 poke36879,25
1520 printchr$(14)
1530 print"▮▮Zustand fuer Tag:"da
1540 print"▮────────────────────"
1550 print"▮Meteor     Entfernung ▮"
1560 fora1=1toa
1565 ifme(a1)<=0andme(a1)=>-10000thenpr
int"Nr."a1tab(11)"▮▮AUFSCHLAG▮":goto158
0
1567 ifme(a1)<-10000thenprint"Nr."a1tab
(11)"▮▮*ABSCHUSS▮":goto1580
1570 print"Nr." tab(3)a1tab(9)"▮"me(a1)
tab(19)"Km"
1580 next
1590 print"▮─────────────────────"
1600 print"▮ges.Bev. ▮"potab(17)"Mill"
1610 print"▮Verluste:▮"vetab(17)"Mill"
1620 print"▮    weiter mit f1▮"
1630 getx$:ifx$<>chr$(133)then1630
1660 print"▮▮Zustand fuer Tag:"da
1670 print"====================="
1680 print"▮Region    Bevoelkerung"
1690 fora1=1to7
1700 print"▮"a$(a1)"▮"tab(10) po(a1)tab
(17)"Mill"
1710 next
1720 print"           ▮Verluste   "
1730 fora1=1to7
1740 print"▮"a$(a1)"▮"tab(10)ve(a1)tab(
17)"Mill"
1750 next
1755 print"▮    weiter mit f1▮"
1760 getx$:ifx$<>chr$(133)then1760
1800 poke36879,76
1805 print"▮▮▮Treffen Sie Ihre Ent- sch
eidung zur Rettung der Erdbevoelkerung!
  "
1810 print
1820 print"<1> medizin.Versorg.        ei
ner Reg."
1830 print"<2> Raketenabschuss         zu
r Vernichtung       eines Meteors"
1840 print"<3> nukleare Vern.          ei
ner Region"
1850 print"<4> Produktion von          Me
dikamenten"
1855 print"<5> Seuchenzustand in       We
ltgrafik"
1860 input"▮▮▮▮Schluesselzahl▮";sz
1870 fort=1to500:next
1900 rem abzweigung
1930 ifsz=1then10000
1940 ifsz=2then20000
1950 ifsz=3then30000
1960 ifsz=4then40000
1965 ifsz=5then50000
1970 ifsz<=0orsz>5then1860
2400 rem var.berech.
2500 fora1=1toa
2505 ifme(a1)=-15000then2550
2510 ifme(a1)>0thenme(a1)=me(a1)-9000
2520 ifme(a1)<=0andme(a1)>-10000thenme(
a1)=0
2530 ifme(a1)<-10000thenme(a1)=-15000
2540 ifme(a1)=0thengosub6000:me(a1)=-10
000:c(b)=1:p1(b)=1
2550 next
2590 ve=0
2600 fora1=1to7
2610 ifc(a1)=0then2640
2620 po(a1)=po(a1)-5:ifpo(a1)<0thenpo(a
1)=0
2625 ve(a1)=px(a1)-po(a1)
2630 ep(a1)=int((1000*ve(a1))/px(a1))/1
0
2640 ve=ve+ve(a1)
2645 ifep(a1)>=100andc(a1)=1thenc(a1+1)
=1:c(a1-1)=1
2800 next
2900 da=da+1:po=px-ve:ifpo<0thenpo=0
2950 ifve=>bythen60000
2960 ifda>=60then60000
4600 fora1=1to13
4610 ifme(a1)<=0andda>30thennext:goto51
00
4700 fora2=1toa
4710 ifme(a2)<=0thennext:goto 600
5000 goto1400
5100 fora1=1to7
5110 ifc(a1)=0thennext:goto60000
5120 goto1400
6000 rem aufschl.ziel
6020 b=int(rnd(1)*9)+1
6030 po(b)=po(b)-2:return
10000 rem med.vers.
10020 poke36879,59:printchr$(144)
10030 print"▮▮      medizinische
   Versorgung        "
10040 print"▮▮verseuchte Regionen"
10050 print
10060 print"▮▮ Region  Nr  Level    "
10065 fora1=1to7:ifc(a1)=0thennext:goto
60500
10070 fora1=1to7
10075 ifc(a1)=0then10100
10080 ifep(a1)<=0then10100
10085 ifve(a1)=px(a1)thenprint"▮"a$(a1)
"▮"tab(9)a1tab(15)" TOT":goto10100
10090 print"▮"a$(a1)"▮"tab(9)a1 tab(13)
ep(a1)tab(19)"%"
10100 next
10110 print">>>>>>>>>>><<<<<<<<<<"
10115 print"MED.VORRAT:"mz"t"
10120 input"▮welche Region▮";a4
10125 print"▮▮noetige Menge:"ve(a4)"t"
10127 ifve(a4)>mzthenprint"▮NICHT GENUG
 VORRAT":goto10140
10130 mz=mz-ve(a4)
10135 print"▮MED.VORRAT:"mz"t"
10137 c(a4)=0
10140 fort=1to3000:next
10150 goto2500
20000 rem raketenab.
20010 poke36879,170:printchr$(28)
20015 print"▮▮▮▮Raketenabschuss"
20030 print"▮▮▮Raketenreichweite =
50000 Km            "
```

Listing »Epedemic« (Fortsetzung)




```
20035 print"■ Meteor    Entfernung  "
20036 ifc(1)>0andc(2)>0andc(3)>0andc(4)
>0andc(5)>0andc(6)>0andc(7)>0then21000
20037 fora1=1toa:ifme(a1)<15000orme(a1)
>50000thennext:goto60300
20040 fora1=1toa
20050 ifme(a1)>50000orme(a1)<15000then2
0120
20060 print"Nr. "a1tab(10)me(a1)"Km"
20120 next
20125 print"**********************"
20130 print"Potential:"rk"Raketen"
20135 ifrk=0then23000
20140 print"■1 Rakete fuer"
20150 input"welchen Meteor";a5
20155 gosub22000
20160 ifa1=1ora1=2then  me(a5)=-15000
20170 a1=0:rk=rk-1
20180 goto2500
21000 print"KEIN ABSCHUSS MOEGLICHDA AL
LE REGIONEN VER- SEUCHT"
21001 print"■DRINGEND MED.HILFE    ERFO
RDERLICH!"
21002 fort=1to8000:next
21005 goto20180
22000 a1=int(rnd(1)*3)+1:ifa1=3thenprin
t"■****ZIEL  VERFEHLT****■":fort=1to500
0:next
22005 return
23000 print"■IHR POTENTIAL IST     ERSC
HOEPFT!"
23010 fort=1to5000:next:goto2500
30000 rem nukl.det.
30020 poke36879,127
30030 print"■■ nukleare Vernichtung"
30035 print"**********************"
30040 print"■um den Befall der     Vire
n auf andere       Regionen zu vermeiden
"
30050 print"ist es noetig eine    Regio
n total zu zer-  stoeren."
30055 print"■■  Region   Nr.  Level  "
30060 fora1=1to7
30070 ifc(a1)=1andep(a1)>50then30080
30075 goto30150
30080 printa$(a1)tab(10)a1tab(14)ep(a1)
tab(19)"%"
30150 next
30160 input"■welche Region";b1
30170 po(b1)=0:c(b1)=0:ve(b1)=px(b1)
30180 goto2500
40000 rem prod.med.
40015 poke36879,158:printchr$(144)
40030 print"■■Medikamente Produktion"
40040 print"Sie Koennen nur in un-verse
uchten Regionen  produzieren!"
40045 print"■■  Region     t Medik. "
40049 aa=0
40050 fora1=1to7
40060 ifc(a1)=1then40120
40070 mk(a1)=int(po(a1)/80)
40075 printa$(a1)tab(13)mk(a1)
40080 aa=aa+mk(a1)
40120 next
40125 mz=mz+aa
40130 print"■■■■■■■■■■■■■■■■■■■■■■"
40140 print"■Med.Vorrat:■"mz"t"
40150 fort=1to5000:next
40160 goto2500
50000 rem weltkarte
50012 poke36879,104:printchr$(142)
50015 print"■■       weltkarte"
50017 forb4=37932to38327:pokeb4,7:next
50019 gosub51000
50020 forq=1to183
50030 readk1
50040 readkj
50050 pokek1,kj
50060 next
50065 poke4558,160:poke4536,160:poke383
50,4:poke38328,7
50066 print"■■■■■■■■■■■■■■■■■■■■■unvers
eucht"
50067 print"■■verseucht"
50070 getx$:ifx$=""then50070
50080 restore
50090 printchr$(14)
50100 goto1800
50500 data4140,108,4141,254,4142,252,41
43,98,4144,98,4145,123,4156,100,4159,11
1,4160,100
50510 data4162,236,4163,160,4164,160,41
65,160,4166,160,4167,160,4168,108,4173,
254
50520 data4174,252,4175,123,4176,108,41
77,254,4178,160,4179,160,4180,160,4181,
160
50530 data4182,160,4183,126,4185,126,41
86,251,4187,160,4188,160,4189,160,4190,1
59000 forc1=38110to38115
59010 pokec1,4:pokec1+c2,4:pokec1+c3,4:
pokec1+c4,4:pokec1+c5,4:pokec1+c6,4:pok
ec1+c7,4
59015 pokec1+c8,4:pokec1+cc,4
59020 next:goto51100
60000 rem************
60010 rem   ende
60020 rem************
60030 poke36879,154
60040 print"■■       Spielende"
60050 ifve=>bythengosub60100
60060 ifve<bythengosub60200
60070 print"Ergebnis:"
60075 print"■ges.Bev.:"po"Mill"
60080 print"■Verluste:"ve"Mill"
60090 print"■Aktionsdauer:"da"Tage"
60095 end
60100 print"■■SIE HABEN DAS SPIEL   VER
LOREN!"
60150 return
60200 print"■■SIE HABEN DAS SPIEL   GEW
ONNEN!"
60250 return
60300 print"■■■Kein Ziel in Aussicht!"
60310 fort=1to2000:next:goto2500
60500 print"■■    zur Zeit keine
  Verseuchung"
60510 print"■■■med.Vorrat:■"mz"t"
60515 fort=1to5000:next:goto2500
READY.
```

Listing »Epedemic« (Schluß)



Der *INTERFACE AGE*

Musik-Synthesizer
für den Commodore 64

EXTENDED SYNTHESIZER SYSTEM ist ein professionelles Musik-Synthesizer-System, das es Ihnen erlaubt, alle Sound-Möglichkeiten des Commodore 64 in vollem Umfang auszunutzen. Sie können fertige Musikstücke abspielen oder neue Kompositionen entwerfen. Die Noten samt aller Zusatzzeichen werden in grafisch hervorragender Weise in allen Details mit der Eingabe oder dem Spielablauf auf dem Bildschirm angezeigt. Alle Möglichkeiten, die es in der Musik gibt, bietet Ihnen EXTENDED SYNTHESIZER SYSTEM in drei Stimmen. Preis: DM 138,– !!!!

Auslieferung EXTENDED SYNTHESIZER SYSTEM für Commodore 64 und Floppy 1541 inklusive ausführlichem deutschen Handbuch mit ca. 50 Seiten.

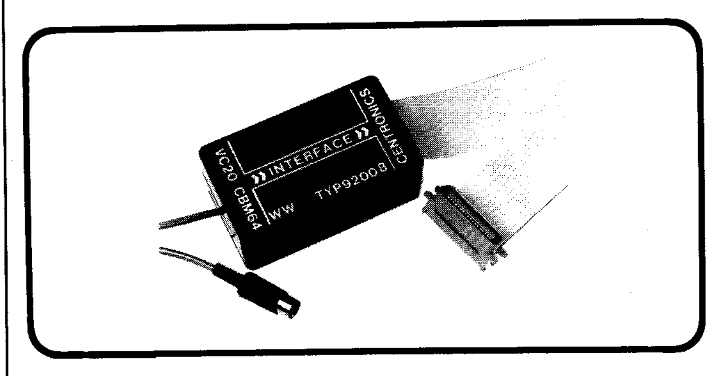
BASIC-COMPILER

- Macht Ihre Programme wesentlich schneller.
- Kompatibel zu EXBASIC LEVEL II, SIMON'S BASIC und SOFTMODULEN.
- Ausführliche 40-seitige Dokumentation.
- Für Commodore 64/1541: DM 298,– inkl. MwSt.
- Für CBM 8032 mit 8050/8250: DM 698,– inkl. MwSt.

INTERFACE AGE Verlag GmbH
Josephsburgstr. 6, 8000 München 80,
Tel. (089) 434089, Telex 5213489 iavmd

Ausführliche Gratisinformationen auf Anfrage.

C 64-Druckerinterface.
Jetzt auch mit 8-KByte-Speicher.



Das neue Druckerinterface Typ 92008 erlaubt, normale Drucker mit Centronics-kompatiblem Eingang an VC 20- und C 64-Computer anzuschließen. Der 8 KByte große Pufferspeicher bewirkt eine deutliche Erhöhung der Arbeitsgeschwindigkeit — Sie können nun gleichzeitig drucken und weiterrechnen. Das Interface ist für 298,— DM inkl. MwSt. sofort lieferbar.

Für NEC-Drucker 8023 und Itoh 8510 ist nun auch ein grafikfähiges Interface zum Anschluß an VC 20 und C 64 lieferbar. Dieses Interface Typ 92000/NEC verfügt über alle CBM-Sonderzeichen und den gleichen Befehlssatz wie CBM-Drucker. Für 298,— DM inkl. MwSt. ist es sofort lieferbar.

Wir liefern Interfaces und Pufferspeicher für fast alle Computer und Peripheriegeräte. Bitte fordern Sie unsere neueste Übersicht oder unseren kompletten Katalog (5,— DM in Briefmarken) an.

Reinhard Wiesemann
Winchenbachstraße 3a
Tel.: 0202/505077

Mikrocomputertechnik
D-5600 Wuppertal 2
Telex: 8591656

COMMANDER
Turbo–Dual–Drive



Das SF 1403 STRING-FLOPPY-SYSTEM ist ein hochentwickeltes Speichermedium, daß dem Anwender neben einem schnellen Datenzugriff und großer Zuverlässigkeit ein günstiges Peripheriegerät bietet, das speziell für den COMMODORE 64 und den VIC 20 konzipiert wurde. Das System basiert auf Endlosband-Basis und arbeitet mit einer Datenübertragungsrate von 19200 Baud pro Sekunde. Weiterhin ist in dem SF 1403 ein intelligenter Controller, ein 4 KB Buffer sowie eine Centronics-Schnittstelle für den Anschluß eines handelsüblichen Druckers und eine zusätzliche RS-232 (serielle) Schnittstelle, die den Anschluß eines seriellen Druckers oder z. B. eines Modems gestattet. Eine außerordentlich zuverlässige Mechanik, der vergleichsweise niedrige Stromverbrauch und die hohe Speicherkapazität machen die SF 1403 STRING-FLOPPY zu einer interessanten Alternative zu einem herkömmlichen Laufwerk.

Spezifikationen:
— Speicherkapazität bis zu 240 KB (formatiert)
— Geschwindigkeit vergleichbar mit Floppy Disk
— kompatibel zu C 64 und VIC 20 sowie sämtlicher Peripherie
— Benutzung von STRING-DRIVES erlaubt, ein Arbeiten mit Files, die größer als der RAM-Bereich sind
— eingebaute Centronics-Schnittstelle
— RS-232 Schnittstelle
— sämtliche Copy- und Back-up-Kommandos möglich
— Maschinensprache LOAD & SAVE
— Völlig unabhängiger I/O
— benutzt lediglich eine Adresse auf dem seriellen Bus



für Spectrum
DM 499,—

für Commodore
DM 599,—

Generalimporteur - Service - Vertrieb
ncs
Nettetaler Computersysteme GmbH
Klemensstr. 7 — D-4054 Nettetal 2 (Kaldenkirchen)
Tel. 02157/1067 Teletex/Telex 215732

Alle Preise verstehen sich incl. 14 % MWSt. Ausführliche Informationen sowie Preislisten und Datenblätter gegen DM 1,30 Rückporto. Händleranfragen ausdrücklich erwünscht.

Reise durch die Wunderwelt der Grafik — Teil 7

In dieser letzten Folge des Grafikkursus wird auf das Hinterschneidungsproblem eingegangen. Außerdem wird eine Lösung für die gleichzeitige Darstellung von hochauflösender Grafik, Text und Mehrfarb-Bit-Map auf einem Bildschirm gezeigt.

Dies ist die letzte Folge unseres Grafikkurses. In der nächsten Ausgabe wird noch ein Maschinenprogramm gezeigt, das die Grafik unterstützt. Das Dornröschen (unsere hochauflösende Grafik) wird dann völlig erwachen und ihre Behäbigkeit wird sich verlieren.

Es stellte sich die Frage, wie man es wohl erreichen könnte, daß nur sichtbare Teile einer Raumfläche gezeichnet werden. H.W. Franke hat in einem Artikel über Computergrafik geschrieben: »Bis heute kann man das Problem noch nicht als völlig abgeschlossen ansehen«. Damit meinte er das Hinterschneidungsproblem. Zumindest für unsere Aufgabenstellung und den C 64 kann man es lösen, wenn man folgendem Gedankengang verfolgt. Stellen Sie sich doch mal den Bildschirm des C 64 in 320 senkrechte Linien unterteilt vor, für jeden möglichen X-Wert also eine Senkrechte. Nun nehmen wir mal eine davon (bei X=X1) und ordnen ihr in unserem Koordinatensystem willkürlich einen Y-Wert 0 zu (Bild 1). In der Doppelschleife (siehe 3D-Programm in der letzten Folge) wird nun irgendwann X=X1 sein und dann der dazugehörige Y-Wert berechnet. Dieser soll zum Beispiel gleich 1 sein. Der Punkt wird gesetzt, und die Schleife läuft weiter (Bild 2).

Wenn alle X-Werte durchlaufen wurden, erhöht sich der Z-Wert in der äußeren Schleife und von neuem wird für X1 ein Y-Wert berechnet. Nehmen wir an, der läge bei 1,5. Auch dieser Punkt wird gesetzt und die Schleife geht weiter (Bild 3).

Wenn nun beim nächsten Schleifendurchlauf X1 erreicht ist, wird es spannend. Es gibt nun drei Möglichkeiten (Bild 4).

In Fall A liegt der neue Punkt oberhalb von P', ist sichtbar und wird gezeichnet. Im Fall B liegt er unterhalb von P, ist ebenfalls sichtbar und wird gezeichnet. Im Fall C aber liegt er zwischen P und P', er ist nicht sichtbar, weil er auf einem uns

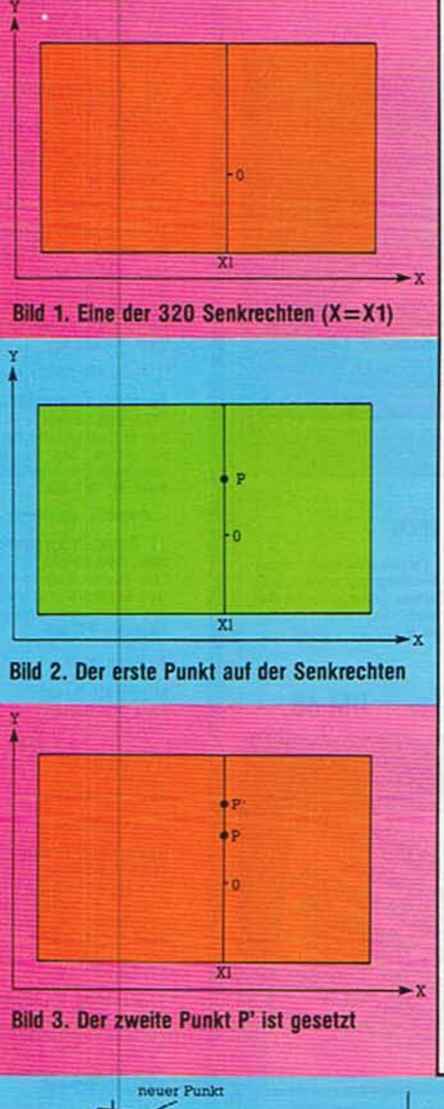


Bild 1. Eine der 320 Senkrechten (X=X1)

Bild 2. Der erste Punkt auf der Senkrechten

Bild 3. Der zweite Punkt P' ist gesetzt

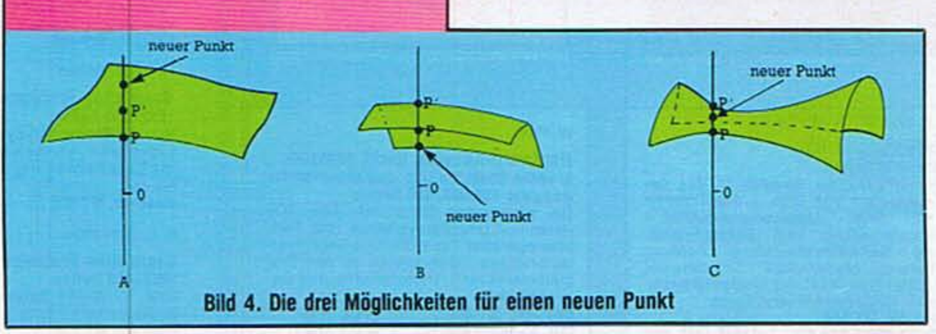


Bild 4. Die drei Möglichkeiten für einen neuen Punkt

abgewandten Hang der Raumfläche liegt und wird daher nicht gezeichnet. Sie sehen also, daß wir uns lediglich immer auf jeder dieser Senkrechten den bisher größten und den bisher kleinsten Wert von Y merken müssen. Bei jedem neuen Y-Wert können wir feststellen, ob er innerhalb des damit aufgespannten Bereiches liegt (dann wird er nicht gezeichnet) oder außerhalb (dann wird er gezeichnet und dieser Wert als der größte oder kleinste bisherige gemerkt). Damit ist für uns das Problem gelöst! Wir richten ein zweidimensionales Feld ein: G (319, 2), wo für jede der 320 Senkrechten drei Werte gespeichert werden können:

G(X,0) als bisher höchster Y-Wert,
G(X,1) als bisher kleinster Y-Wert,
G(X,2) als Kennmarke, ob für diesen X-Wert schon ein Y-Wert aufgetaucht ist (in dem Fall ist G(X,2)>0) oder noch nicht (dann ist G(X,2)=0).

Durch das RUN-Kommando sind alle Variablen gleich 0, also auch G(X,2). Genau besehen benötigen wir diese Kennmarke, ob auf der Senkrechten durch X schon ein Punkt gesetzt wurde, nicht unbedingt in unserem Programm. Es ist so gestaltet, daß beim X-Durchlauf keine Lücken gelassen werden. Sinnvoll ist es trotzdem, sie einzurichten, denn durch die 45°-Verschiebung erhöht sich ja der maximale X-Wert ständig und außerdem könnte man sich ja mal überlegen, ob man den X-Durchlauf mit Lücken macht. In der letzten Folge hatten wir im Flußdiagramm unsere Doppelschleife entwickelt. Dabei war ein Teil »Zeichnen des Punktes X,Y«. Diesen Teil ersetzen wir durch den in Bild 5.

Jetzt sehen wir uns das als Programmteil an. Also Computer anschalten, Laden des 3D-Programmes aus der letzten Folge sowie der Grafik-Unterprogramme aus der vierten Folge (entweder — falls Sie das haben — durch MERGEn oder als kombiniertes Programm aus der Arbeit in der letzten Folge). Zeile 230 des 3D-Programmes enthält als letzten Befehl GOSUB 50040, den Aufruf

zum »Punkt zeichnen«. Den löschen wir jetzt und fügen die folgenden Zeilen ein:

```
232 IF G(X,2)=0 THEN G(X,0)=
Y:G(X,1)=Y:GOTO 238
234 IF Y>G(X,0) THEN G(X,0)=
Y:GOTO 238
236 IF Y<G(X,1) THEN G(X,1)=
Y:GOTO 238
237 GOTO 240
238 G(X,2)=G(X,2)+1:GOSUB 50040
```

Außerdem muß natürlich zu Beginn dieses Feld noch dimensioniert werden: 147 DIM G (319,2)

Mit der Beispielfunktion Y=COS(Z) *SIN(X) und den Eingaben:XU=-1, XO=10, YU=-2, YO=5, ZU=-.1, ZO=7, Schrittweite=8,XA=0, XE=6, ZA=.1, ZE=7 sieht man den Effekt jetzt ganz deutlich. Das einzige — außer der Geschwindigkeit (und naja-vielleicht noch ein paar Kleinigkeiten) — was unsere 3D-Grafik jetzt von professionellen Systemen unterscheidet, ist die Möglichkeit der Netzgrafik (Bild 6).

Dieses Thema soll nicht ausführlich behandelt werden, sondern wir werden nur zwei Wege zur Netzgrafik anschauen:

Weg 1:
Wir machen die Schrittweite in Z-Richtung sehr klein (so wie die in X-Richtung), setzen aber nicht jeden Punkt, sondern zum Beispiel in folgender Anordnung:

a) 1. Z-Wert: Jeder Punkt der X-Schleife wird gesetzt, dann
b) 2. bis 7. Z-Wert: Nur jeder 8.Punkt der X-Schleife wird gesetzt, dann
c) wieder weiter wie beim 1. Z-Wert und so weiter.

Stark vergrößert hätten wir dann etwa Verhältnisse wie in Bild 7.

Diese Lösung ist programmtechnisch einfacher als Weg 2 und ganz gut an unser bisheriges 3D-Programm anzuhängen. Nun aber noch zu

Weg 2:
Man läßt außer der Z-X-Doppelschleife zur Berechnung von Y auch noch eine X-Z-Doppelschleife laufen. Wenn Sie das ausprobieren, gibt es allerdings wieder Schwierigkeiten mit den nicht sichtbaren Linien, denn unser Verfahren zur Lösung des Hinterschneidungsproblems läßt nichts mehr zeichnen, was innerhalb von G(X,0) bis G(X,1) liegt!

Anscheinend hat man bisher noch keine Möglichkeit Erklärungen auf das Grafikbild zu schreiben. Wenn wir im Hochauflösungsmodus zum Beispiel das Programm durch <RUN/STOP> anhalten, dann tauchen alle Meldungen als farbige Quadrate auf dem Bildschirm auf. Der Inhalt des Bildschirmspeichers dient ja jetzt als Farbinformation. Was der Computer mit dieser Farbkombination dann jeweils auf dem Bildschirm zeigen soll, holt er sich aus der Bit-Map. Man kann natürlich ohne weiteres die wichtigsten Zeichen aus Punkten, Linien und Ellipsenbögen zusammenbauen unter Verwendung unserer Grafik-Unterprogramme. Das wäre sozusagen der »harte« Weg. Aber wozu haben wir im Speicher schon die fertigen Zeichenmuster liegen! Wir müßten nur auf sinnvolle Weise an sie herankommen. Prinzipiell gibt es zwei »weiche« Wege:

a) Herbert Kunz hat den einen davon in der Zeitschrift Computer persönlich, Ausgabe 2 (1984), Seite 78 vorgestellt. Er kopiert zunächst den Zeichensatz in einen RAM-Bereich, schaltet dann — wie gewohnt — in den Hochauflösungsmodus (bei ihm liegt der Bildschirm bei 1024 und die Bit-Map richtet er bei 8192 ein) und druckt den Text auf den Bildschirm, wo dieser erst mal in farbigen Quadraten auftaucht. Nun sieht er mittels PEEK nach, welches Zeichen an der Bildschirmstelle steht (wo jetzt natürlich nur ein farbiges Quadrat zu sehen ist). Was Herbert Kunz dadurch erhält, ist die Kennzahl (der Bildschirmcode, zum Beispiel für ein A eine 1), die uns sagt, an welcher Stelle der Zeichentabelle das gefragte Zeichen steht. Dabei ist allerdings zu bedenken, daß es auch eine nullte Stelle gibt. Jedes Zeichen besteht aus 8 Bytes und deshalb multipliziert er die Kennzahl mit 8 und addiert sie zur Startadresse des RAM-Bereiches hinzu, in den er das Zeichen-ROM kopiert hat. Von da an überträgt er Byte für Byte das Zeichen in die entsprechende Stelle der Bit-Map (die er aus Zeilen und Spaltenangabe berechnet). So wird es dann sichtbar.

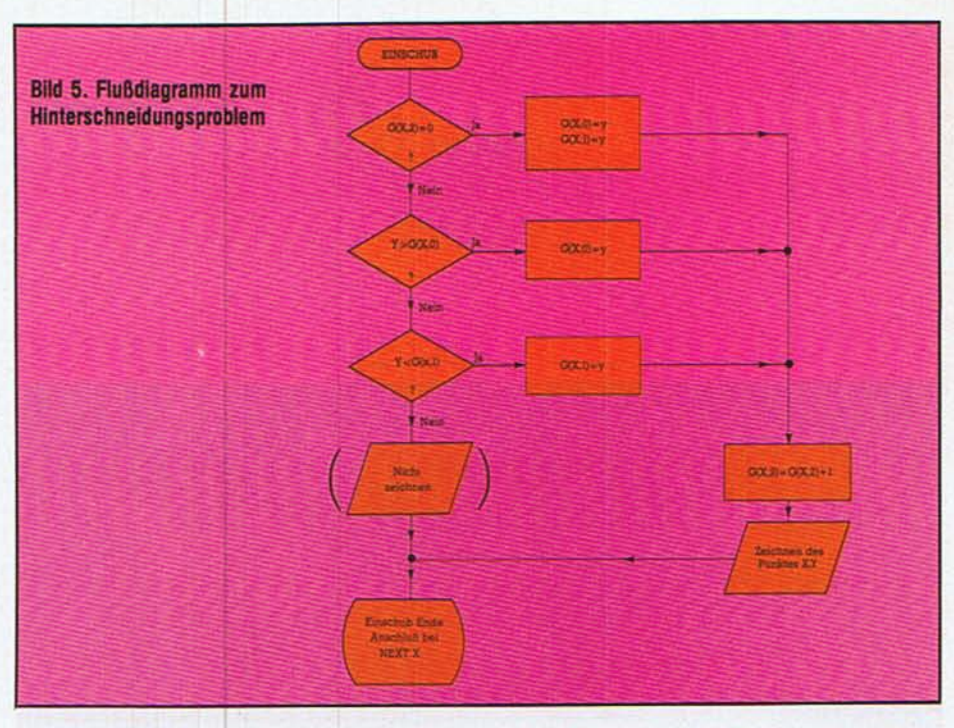


Bild 5. Flußdiagramm zum Hinterschneidungsproblem

Das macht er Zeichen für Zeichen bis der gesamte — in einem String definierte — Text in der Bit-Map und damit für uns lesbar auf dem Bildschirm steht. Dieses Prinzip können wir in unsere Programme übernehmen. Dazu sind nur wenige Änderungen nötig. Zunächst schließen wir den neuen Zeichen-Speicher im RAM direkt an unsere Bit-Map an: ab 32768. Das Maschinenprogramm von Herbert Kunz verändern wir deshalb etwas und legen es nicht in den Kassettenpuffer, sondern von dez. 673 bis 715. Wie im Programm von Herbert Kunz brauchen wir dann nur noch die Angaben, in welcher Zeile und Spalte welcher Text geschrieben werden soll.

Zunächst einmal wird das Maschinenprogramm eingelesen und ausgeführt. In das 2D- und in das 3D-Programm fügen wir ein:

```
500 FOR I=673 TO 715: READ A:PO-
KE I, A:NEXTI:SYS 673: RETURN
510 DATA 120, 169, 49, 133, 1, 169, 0,
133, 98, 133, 100, 169, 208, 133, 99, 169,
128, 133
520 DATA 101, 162, 16, 160, 0, 177, 98,
145, 100, 200, 208, 249, 230, 99, 230,
101, 202, 208
530 DATA 242, 169, 55, 133, 1, 88, 96
```

In Zeile 5 hängen wir an die beiden POKE-Kommandos noch an :GOSUB 500

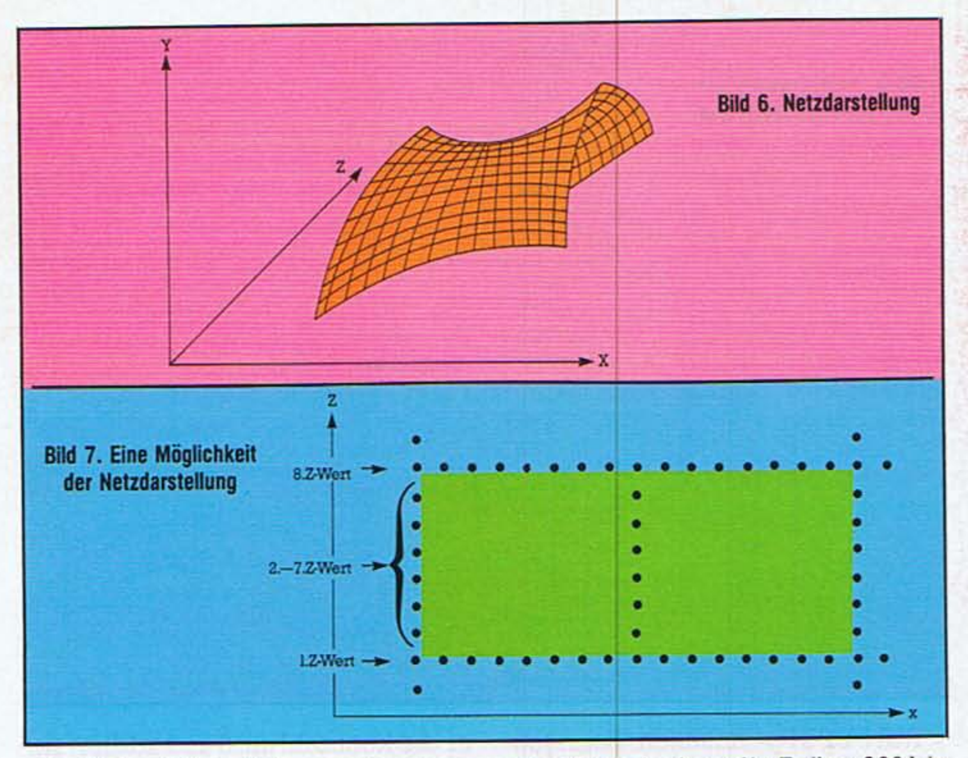


Bild 6. Netzdarstellung

Bild 7. Eine Möglichkeit der Netzdarstellung

In Zeile 120 definieren wir einen String, der einen senkrechten Tabulator ermöglicht:
120 CP$=CHR$ (19):FOR I=1 TO 24:CP$=CP$+CHR$ (17):NEXT I
Dann definieren wir den zu druckenden String und Zeile und Spalte des Druckortes:
122 TE$="Y"+F$:ZE=0:SP=0

Jetzt müssen wir nur noch dafür sorgen, daß nach dem Zeichnen der Kurve oder Raumfläche der Text gedruckt wird. Dazu gleichen wir zunächst mal das 2D- an das 3D-Programm an. In den Zeilen 200, 210, 220 des 2D-Programmes ändern wir die Zeilennummern zu 300, 310, 320 (dann muß natürlich in Zeile 300 die Anweisung THEN 200 zu THEN 300 umgeschrieben werden und die alten Zeilen 200 bis 220 gelöscht werden). Im 2D-Programm hört das Zeichnen in Zeile 195, im 3D-Programm in Zeile 250 auf. Deswegen legen wir den Druckvorgang ab Zeile 260:
260 PRINT LEFT$ (CP$,ZE+1)TAB (SP)TE$;
262 AN=23552+ZE*40+SP:GS=24576+ZE*320+SP*8
264 FOR I=AN TO AN+LEN (TE$)-1
266 L=PEEK (I) :Z=32768+8*L:POKE I, 16*F1+F2
268 FOR J=O TO 7:POKE GS+J, PEEK (Z+J):NEXT J
270 GS=GS+8:NEXT I

Wenn Sie das Programm (2D- oder 3D-Programm) jetzt starten, bekommen Sie als Kopf des Hochauflösungsbildschirmes noch die gezeichnete Funktion als Gleichung gedruckt. Sie können sich leicht aus diesen Angaben ein Programm schreiben, in dem die Zeilen 260 bis 270 ein Unterprogramm bilden, das jeweils mit neuem Text TE$, neuer Zeile ZE und Spalte SP aufgerufen werden kann. So können Sie beliebigen Text in das Hochauflösungsbild schreiben.

b) Nun zum zweiten »weichen« Weg. Weil sich dieser nahezu völlig in Maschinensprache abspielt, soll hier nur das Prinzip erklärt werden. In der Serie über Assembler-Programmierung wird dieses sogenannte Interrupt-Handling besprochen. Zunächst das Programm. Tippen Sie NEW ein und dann Listing 1.

Starten Sie mit RUN und Ihr Bildschirm ist in drei Zonen aufgeteilt. Ist das Programm beendet, drücken Sie eine Taste. Sie sind auch nach dem END noch in dieser Bildschirm-Aufteilung. Sie können das leicht feststellen, wenn Sie den Cursor nach oben bewegen und ein paar Schreibversuche machen. Oder versuchen Sie mal das LIST-Kommando! Wenn Sie davon genug haben, dann geben Sie (RUN/STOP) und (RESTORE) ein, und Sie sind wieder im Normalzustand des Bildschirmes. Wenn man diese Technik beherrscht, kann man mit dem Bildschirm praktisch alles machen, was man will! Wir bedienen uns eines sogenannten Rasterzeilen-Interrupts. Ich habe Ihnen in der letzten Folge erklärt, daß der bildaufbauende Elektronenstrahl über zirka 280 Rasterzeilen huscht und 20mal in der Sekunde ein komplettes neues Bild aufbaut. In der einschlägigen Literatur ist man sich übrigens uneins: Woanders wird erzählt, es handle sich um 264 Rasterzeilen und 60 mal in der Sekunde werde ein neues Bild aufgebaut. Wie dem auch sei: Richtig ist, daß es ein enorm schneller Geselle ist, der über den Bildschirm huscht, und daß die aktuelle Rasterzeile in den Registern 53266 (LSB) und 53265, Bit7 (msb) mitgezählt wird. Interrupts bringen den Computer dazu, neben seiner uns sichtbaren Arbeit (zum Beispiel Programmablauf) noch eine Anzahl anderer Dinge zu tun. Eines davon ist die ständige Wiederauffrischung des Fernsehbildes durch Informationen an den Rasterstrahl. Der C 64 löst solche Interrupts auch per Programm aus. Zu diesem Zweck dienen die Register 53273 und 53274. Bit 7 von 53273 sagt uns, daß ein Interrupt aufgetreten ist (Bit 7 ist dann 1), mit einer der Ursachen, die noch in den Bits 0 bis 3 einzeln angegeben werden:
Bit 0=1 Rasterzeilen-Interrupt
Bit 1=1 Sprite/Hintergrund-Kollision
Bit 2=1 Sprite/Sprite-Kollision
Bit 3=1 Interrupt durch Lichtgriffel.

Register 53274 bietet uns die Möglichkeit einer sogenannten Interrupt-Maske. Bis auf Bit 7 ist es genauso aufgebaut wie 53273. Wenn wir hier zum Beispiel in Bit 0 eine Eins setzen, dann weiß der VIC-II-Chip, daß er einen sogenannten Rasterzeilen-Interrupt auslösen soll. Nur dies alleine würde kaum Wirkung haben, denn nun erfolgt beim Auslösen des Interrupts ein Sprung an die Adresse, die vom Interrupt-Vektor (Speicherstellen 788 (LSB) und 789 (MSB)) angezeigt wird. Das ist im Normalfall ein Maschinenprogramm im Betriebssystem (Start bei 59963). Weil dieser Zeiger im RAM liegt, kann er verändert werden, so daß er auf ein eigenes Maschinenprogramm weist, das nun die Interruptbehandlung nach unserem Gutdünken ausführt. Außerdem müssen wir noch angeben, in welcher Rasterzeile der Interrupt stattfinden soll. Dazu schreiben wir in das Rasterregister diese Zeilennummer ein. Das ist also das Prinzip, und weil diese ganze Angelegenheit sehr schnell erledigt sein muß, ist das nur in Maschinensprache möglich.

Grafik und Maschinensprache

Wer Grafik in Basic betreibt, braucht Sitzfleisch. Das haben Sie sicherlich am eigenen Leibe bemerkt. 20 Minuten für ein fertiges

3D-Bild, das ist schon ziemlich lange! Die meisten brauchbaren Grafik-Programme sind deshalb in Maschinensprache geschrieben. In der nächsten Ausgabe wird ein einfaches aber brauchbares Grafik-Unterprogramm-Paket in Maschinensprache vorgestellt, mit dem Sie den Zeitbedarf erheblich reduzieren können. Verantwortlich für diese lange Zeitdauer von Basic-Programmen ist der Basic-Interpreter, der jeden Befehl übersetzen muß und dann ein zum Befehl gehöriges, oft recht verwickeltes Maschinenprogramm ausführt, dann den nächsten Befehl übersetzt, und so weiter... Wenn das zum Beispiel in einer FOR....NEXT-Schleife mit 320 Durchläufen passiert, dann dauert das...! Je einfacher und auch allgemein verwendbarer ein Maschinenprogramm ist (jedenfalls für Grafik), desto umfangreicher muß das Basic-Aufrufprogramm sein. Oder: Je spezialisierter ein Maschinenprogramm ist, desto weniger Basic-Aufrufprogramm ist nötig. Ein Beispiel: Wenn das Maschinenprogramm lediglich die Routine zum Berechnen und Zeichnen eines Punktes enthält, dann muß vom Basic-Aufrufprogramm eine FOR..NEXT-Schleife 320 durchlaufen werden, in der jedesmal X variiert, Y aus der Funktionsgleichung berechnet wird, beide dann transformiert werden auf das Bildschirmsystem, die transformierten Werte an die Speicheradressen gePOKEt werden, von denen es die Maschinenroutine nach dem SYS-Aufruf abholt. Man hat im Aufrufprogramm allerlei Freiheiten: Man kann nach Belieben das Koordinatensystem ändern, die zu zeichnende Funktion, den Start- und den Endwert der FOR..NEXT-Schleife und das alles relativ einfach durch einige INPUT-Anweisungen oder notfalls Programmzeilenänderungen erreichen. Dafür muß man die lange Zeitdauer des Aufrufprogrammes hinnehmen. Ein Maschinenprogramm wäre zwar sehr schnell und erforderte von Basic aus unter Umständen nur einen SYS-Befehl. Aber es wäre mehr oder weniger festgelegt auf immer diese eine Aufgabe und damit recht unbeweglich und auch ziemlich lang. Maschinenprogramme andererseits, die dieselben Eingabe- und Variationsmöglichkeiten wie das vorhin erwähnte Basic-Aufrufprogramm bieten, sind schon etwas für Feinschmecker der Assemblerprogrammierung und äußert rar. Ein anderes Phänomen ist die Tatsache, daß man vor der Wahl steht: Geschwindigkeit oder Speicherplatz sparen. Man kann Grafik-Maschinenprogramme enorm beschleunigen durch spezielle Programm-Techniken. Sie werden dann aber häufig so lang, daß man sie kaum mehr als DATA-Zeilen-Listing abdrucken kann.

```
5 REM***GRAFIK,TEXT UND MEHRFARBGRAFIK**
7 REM*EINLESEN UND ANSCHALTEN ML-PROGR**
10 FORI=49152TO49270:READA:POKEI,A:NEXT:SYS49152
15 REM***** ERLAEUTERUNGEN DRUCKEN ******
20 PRINTCHR$(147):FORI=0TO8:PRINT:NEXT
30 PRINT"OBEN:HOCHAUFLOESUNGS-MODUS"
40 PRINT:PRINT"MITTE:NORMALER TEXT-MODUS"
50 PRINT:PRINT"UNTEN:MEHRFARBEN-BIT-MAP-MODUS"
55 REM++FARBEN SETZEN,BIT-MAP LOESCHEN++
60 FORI=1024 TO1383:POKEI,114:NEXTI:FORI=1384TO1423:POKEI,0:NEXTI
70 FORI=1664TO2023:POKEI,234:NEXTI
80 FORI=55936TO56295:POKEI,13:NEXTI
90 FORI=8192TO11391:POKEI,0:POKEI+4800,0:NEXT
95 REM+++++++++ KURVEN ZEICHNEN+++++++++
100 BA=8192
105 REM+++++HOCHAUFLOESUNGSMODUS+++++++
110 A=40:B=0:FORX=0TO319:GOSUB150:NEXTX
115 REM++++MEHRFARBEN BIT-MAP-MODUS+++++
120 A=160:B=0:FORX=0TO319STEP2:GOSUB150:NEXTX
125 B=40:FORX=1TO319STEP2:GOSUB150:NEXTX
130 B=80:FORX=0TO319STEP2:C=0:GOSUB150:C=1:GOSUB150:NEXTX
135 REM+++++++++++ABSCHALTEN+++++++++++
140 GETA$:IFA$=""THEN140
144 END
145 REM +++++++ FUNKTION (SINUS) +++++++
150 Y=INT(A+20*SIN(X/10+B))
155 REM +++PUNKTE BERECHNEN,ZEICHNEN++++
160 BY=(XAND504)+40*(YAND248)+(YAND7):BI=ABS(7-(XAND7)-C)
170 POKEBA+BY,PEEK(BA+BY)OR(2↑BI):RETURN
180 REM++++MASCHINENPROGRAMM+++++++++++
200 DATA120,169,127,141,13,220,169,1,141,26,208,169,3,133,251,173,112,192
201 DATA141,18,208,169,24,141,17,208,173,20,3,141,110,192,173,21,3,141,111
202 DATA192,169,50,141,20,3,169,192,141,21,3,88,96,173,25,208,141,25,208
203 DATA41,1,240,43,198,251,16,4,169,2,133,251,166,251,189,115,192,141,33
204 DATA208,189,118,192,141,17,208,189,121,192,141,22,208,189,124,192,141
205 DATA24,208,189,112,192,141,18,208,138,240,6,104,168,104,170,104,64,76
206 DATA49,234,49,170,129,0,6,0,59,27,59,24,8,8,24,20,24
READY.
```

Listing 1. Gleichzeitige Darstellung von Grafik, Text und Mehrfarb-Bit-Map.

Wie man einen Trickfilm mit Hilfe von Sprites drehen kann, habe ich Ihnen in Folge 5 gezeigt. Welche Möglichkeiten gibt es ohne Sprites? Schneller Bildaufbau — kurze Verzögerung — Bild löschen — neues Bild aufbauen mit veränderter Ansicht — und so weiter... Wenn Sie das in Basic überlegen, können Sie das gleich wieder vergessen. Um den Eindruck von Bewegung zu vermitteln, muß alles viel schneller gehen. Aber auch ein Maschinenprogramm muß sehr sorgfältig entwickelt werden, um die nötige Geschwindigkeit des Bildaufbaues zu erhalten. Für einfache Darstellungen könnte dieses Konzept aber funktionieren. Denken Sie aber zum Beispiel mal an unsere 3D-Bilder! Mit dem C 64 — behaupte ich — geht's so nicht. Stellen Sie sich vor, wir benützen mehrere Bit-Maps und

Bewegte Grafik

Bildschirme und zeichnen in je eine Bit-Map einen Bewegungszustand unseres Bildes. Dann müssen wir nur noch in einer Aufrufschleife von Bit-Map zu Bit-Map umschalten. Das ist auch in Basic möglich. Wir hätten dann drei Bit-Maps zur Verfügung. Das ist zwar nicht viel, aber immerhin könnte man damit schon eine Raumfläche in zwei Richtungen kippen oder ähnliches. Wenn man in Maschinensprache programmiert, hat man sogar 5 Bit-Maps (mindestens) zur Verfügung und weil auch noch alles schneller geht, kann man mit einigem Geschick vielleicht sogar während man vier Bilder nacheinander zeigt, das fünfte Bild ungesehen aufbauen und auf diese Weise mehr als fünf Bewegungsstadien realisieren. Sie sehen aber schon: Das ist die hohe Schule der Programmierkunst und hier gibt es noch viel zu tun. Einen 3D-Trickfilm auf diese Weise zu erzeugen, ist heute auch auf großen Computern noch kaum möglich. Der Film TRON beispielsweise setzt sich aus lauter per Computer erzeugten Einzelbildern zusammen, die erst filmtechnisch zum Bewegungsablauf aneinandergehängt wurden. Dort wo Echtzeit-Darstellung notwendig ist, bei Simulationen beispielsweise in Flugtrainern, reduziert man die Darstellung auf das unbedingt notwendige und hat außerdem dazu Computer zur Verfügung, die uns C 64-Benutzern das Wasser im Mund zusammenlaufen lassen. Aber was solls. Die Kunst des Programmierens liegt ja vielleicht darin, daß man mit einem Minimum an Aufwand einen Maximaleffekt erzielt. Und wenn man sich dann mal ansieht, was wir aus unserem C 64 alles herausholen können, dann stehen wir eigentlich ganz gut da, meinen Sie das nicht auch?

(Heimo Ponnath/aa)

Assembler ist keine Alchimie

Bisher haben wir uns mit dem Innenleben unserer Computer auseinandergesetzt und die wichtigsten Teile der Hardware kennengelernt. Jetzt kommen wir wie versprochen zur Software, nämlich zum Assembler.

Sollten Sie einen Assembler haben, dann schalten Sie ihn jetzt bitte ein. Falls Sie keinen haben, schaffen wir Abhilfe. Wir werden in den nächsten Ausgaben ein komplettes Software-Paket; Assembler, Disassembler und Monitor veröffentlichen. Dieser Assembler wird uns dann den ganzen Kurs hindurch begleiten. Meistens meldet der Assembler sich mit einer Registeranzeige. Sollte Ihrer das nicht tun, dann müssen Sie wohl noch den Monitor anschalten oder speziell einen Befehl für die Registeranzeige eingeben (häufig ist das ein R). Jedenfalls wird nun auf dem Bildschirm der Inhalt der Register angezeigt, (Bild 1).

Die angezeigten Werte sind Beispiele, wie sie beim C 64 auftreten können. PC ist der Programmzähler, der immer auf den nächsten zu holenden Befehl zeigt. (Der Wert $E147 rührt vom SYS-Aufruf, mit dem ich meinen Assembler starte). IRQ zeigt uns an, auf welche Adresse der sogenannte Interrupt-Vektor gestellt ist. Das ist das Byte-Paar 788 (LSB) und 789 (MSB). Auf den Wert $EA31 zeigt es im Normalfall.

Die nächsten acht Angaben beziehen sich auf das Prozessorstatusregister, das wir in der letzten Folge P genannt haben. Die Bedeutung der einzelnen »Flaggen« zeigt Ihnen (Bild 2).

AC ist der aktuelle Inhalt des Akkus.

XR zeigt an, was im X-Register und YR was im Y-Register enthalten ist.

SP (von Stack-pointer = Stapelzeiger) gibt uns Auskunft über den freien Platz im Stapelregister. Damit wissen wir genau, was in diesem Moment in unserem Computer vorgeht. So fremd Ihnen das alles im Augenblick noch vorkommt, bald werden Sie mit dieser Registeranzeige auf vertrautem Fuß stehen.

Wie sieht ein Assemblerprogramm aus?

Das menschliche Gehirn hat dem des Computers vieles voraus. Dazu gehört es beispielsweise, daß ein Mensch allerlei Dinge gleichzeitig tun kann: gehen, sprechen, Musik hören, lächeln, Handbewegungen vollführen, womöglich dabei auch noch etwas kauen und so weiter. Ein Computer ist dazu nicht imstande. Er erledigt eine kleine Aufgabe nach der anderen. Weil er das so schnell macht, hat es für uns den Anschein, es geschähe alles gleichzeitig. Das Maschinenprogramm ist eine Kette solcher kleiner Aufgaben. Das erste Glied daraus, das wir kennenlernen wollen ist der Befehl

LDA.

Das bedeutet: Lade den Akkumulator. Alle Assembler-Befehlsworte bestehen aus drei Buchstaben wie dieser hier auch. Wir haben in der ersten Folge schon gesagt, daß einem solchen Befehl eine 8-Bit-Codezahl entspricht. Das ist hier $A9 oder binär 1010 1001 oder schließlich dezimal 169. Die Code-Zahl muß in einem Speicherplatz stehen, zum Beispiel in $1500 (entspricht dez. 5376). Assemblerlistings sehen dann so aus:

1500 LDA

Hier tritt also die Speicherplatznummer mit einem nachfolgenden Befehl anstelle der vom Basic gewohnten Zeilennummer.

Hier fehlt noch etwas Entscheidendes: Was soll den in den Akku geladen werden? Genauso wie es in Basic Befehle gibt, die für sich alleine stehen können wie CLR oder LIST, gibt es auch im Assembler solche Befehle. Weitaus häufiger aber sind hier Befehle, die ein Argument erfordern (in Basic zum Beispiel PEEK(100)). Dabei ist 100 das Argument). In Assembler gibt es zwei Sorten von Argumenten. Solche, die in einem Speicherplatz unterzubringen sind und andere, die zwei Bytes brauchen. Mit dem Befehlswort (hier also LDA) zusammen gezählt, existieren in Assembler also 1-Byte-Befehle, 2-Byte-Befehle und 3-Byte-Befehle.

Das Argument von LDA ist also das, was in den Akku soll. Laden wir also mal eine 1 in den Akku:

1500 LDA #$01

Wir haben jetzt einen 2-Byte-Befehl erzeugt. Was aber bedeuten # und $ dabei? $ ist leicht zu erklären. Die große Mehrzahl der Assembler nimmt bei Zahlenangaben Hexadezimalzahlen an. Bei einigen muß man das durch das $-Zeichen kennzeichnen. Manche Assembler lassen auch Binärzahlen, Dezimalzahlen und sogar ASCII-Zeichen als Argumente zu. Für jede Eingabeart steht dann vor dem Argument ein Zeichen, das die Art des Argumentes angibt, zum Beispiel häufig »!« für Dezimalzahlen oder % für Binärzahlen. Nun zum #-Zeichen. Es gibt viele Arten, den Akku zu laden. Direkt mit einer Zahl — wie wir hier —, aber zum Beispiel auch mit dem Inhalt eines anderen Speichers und so weiter. Man spricht von der sogenannten Adressierung.

Es gibt eine ganze Menge davon und jede wird auf eindeutige Weise gekennzeichnet. Wenn wir in unseren Akku eine Zahl laden, dann ist das die »unmittelbare« Adressierung, und die kennzeichnet man mit dem #-Zeichen.

Wenn in Speicherstelle $1500 die Codezahl für LDA steht, dann muß die 1 in der Speicherstelle $1501 stehen, wie es sich für einen 2-Byte-Befehl gehört. Wenn Sie nun die Assemblerzeile eingegeben haben und (RETURN) drücken, dann taucht auf dem Bildschirm ... eine Fehlermeldung auf (bei vielen Assemblern). Wir müssen vorher nämlich noch unserem Software-Instrument sagen, jetzt zu assemblieren. Wie das geschieht, ist auch wieder von Assembler zu Assembler verschieden. Die meisten erwarten, daß man vor der Zeile noch ein A eingibt:

A 1500 LDA #$01

Wenn Sie jetzt (RETURN) drücken, zeigt der Bildschirm:

A 1500 LDA #$01 A 1502

und meistens einen blinkenden Cursor, der auf die nächste Eingabe wartet. $ 1502 ist die nächste freie Speicherstelle, und wenn beim Programmablauf der Programmzähler nach dem LDA #$01 auf $1502 deutet, dann erwartet er dort den nächsten Befehl. Wenn dort Unsinn steht, dann stürzt der Computer im allgemeinen ab, je nachdem, welcher Code dann hier zufällig enthalten ist. Wir haben ja 256 Möglichkeiten dafür: $00 bis $FF. Im Gegensatz zu Basic, wo man durch den Interpreter die Möglichkeit hat, Zeilennummern zu bauen wie man will, muß hier das Progamm eine ununterbrochene Perlenschnur von Befehlen in Speicherstellen sein. Durch einige

Befehle läßt sich dieses Prinzip allerdings durchbrechen.

Damit wir die Wirkung von Befehlen sehen können, greife ich auf einen Befehl vor, der ähnlich dem STOP in Basic einen Programmabbruch bewirkt: BRK. Die genaue Funktion soll erst später erklärt werden, aber wir sehen jedenfalls dann, wenn ein Maschinenprogramm auf einen BRK-Befehl läuft, die Registerinhalte angezeigt. Das ist in den meisten Assemblern eingebaut. Wir ergänzen jetzt:

A 1502 BRK

Damit erstmal genug. Steigen Sie aus dem Assembler aus, und starten Sie das Programm. In den meisten Assemblern geht das mit

G 1500

oder sonst von Basic aus mit SYS 5376. Jetzt werden wieder die Register angezeigt. Der Programmzähler steht auf 1503, im Akku steht 01, alle Flaggen außer der Breakflagge sind Null (die unbenutzte Flagge steht immer auf 1). Jetzt ändern wir das Argument:

A 1500 LDA #$00
A 1502 BRK

Wir starten wieder und sehen uns die Register an: Programmzähler 1503, Akku jetzt 00, aber bei den Flaggen hat sich etwas verändert: Die Zero-Flagge ist auf 1 gesetzt. Wir sehen also: Diese Flagge bleibt solange ungesetzt, solange nicht eine Null im Akku auftaucht, erst dann wird sie 1.

Noch einmal ändern wir das Programm:

A 1500 LDA #$FF
A 1502 BRK

Nach erneutem Start steht das Erwartete in den Registern, nur bei den Flaggen ist etwas Merkwürdiges passiert: Die Vorzeichenflagge steht auf 1. Das bedeutet, im Akku soll eine negative Zahl stehen! Nun wissen wir aber, daß $FF = dez. 255 ist. Dieses Rätsel wird uns noch eine Weile begleiten. Es sei hier nur bemerkt, daß kein Fehler vorliegt: Immer wenn in einer Zahl das Bit 7 gleich 1 ist, geht die Vorzeichenflagge auf 1. Die Lösung des Rätsels werden wir bei den negativen Binärzahlen finden.

Wir schließen aus alledem: Der LDA-Befehl beeinflußt die Vorzeichen- und die Zeroflagge.

Der zweite Assembler-befehl: STA

STA heißt »store accumulator«, also »lege Akkuinhalt ab«. Wie Sie sich denken können, muß auch hier ein Argument auftauchen: Nämlich wohin abgelegt werden soll. Wir legen unseren Akkuinhalt in die erste Bildschirmspeicherstelle (C 64:$0400, VC 20 Grundversion: $1E00, VC 20 mit Erweiterung: $1000). Unser Programm muß also so aussehen:

A 1500 LDA #$01
A 1502 STA $0400 oder die entsprechende Adresse (siehe oben).

Mit diesem STA-Befehl lernen wir eine neue Adressierungsart kennen: Die »absolute« Adressierung. Sie ist daran zu erkennen, daß kein besonderes Merkmal verwendet wird. Die Adresse $ 0400 ist nicht in einem Byte darstellbar, sondern wird aufgeteilt auf zwei Bytes. Im Speicher steht jetzt:

1500 LDA #
1501 $ 01
1502 STA
1503 $ 00 »das ist das LSB«
1504 $ 04 »das ist das MSB«

Hier liegt also ein 3-Byte-Befehl vor, und die nächste freie Speicherstelle ist $ 1505.

Vom Basic her wissen Sie, daß 1 der Bildschirmcode für den Buchstaben A ist, und daß man jeder Bildschirmspeicherstelle auch eine Bildschirmfarbspeicherstelle zuordnet. Um ein eingeschriebenes Zeichen vom Hintergrund abzuheben, muß man dort dann eine Farbinformation eingeben. Der Start dieses Bildschirmspeichers liegt so:

C 64 :$ D800
VC 20 (Grundv.): $9400
VC 20 (Erw. Vers.): $9600.

Der Farbe Schwarz entspricht die Code-Zahl 0. Wir ergänzen unser Programm durch:

A 1505 LDA #$00
A 1507 STA $D800 (oder entsprechender Speicher, siehe oben). Die nächste freie Adresse ist nun $150A.

Unser Programm soll jetzt abgeschlossen sein. Damit der Computer aber beim Programmzählerstand $150A nicht Unsinn vorfindet, muß — ähnlich wie bei END in Basic — das Programm auf irgendeine Weise beendet werden. Das kann durch BRK geschehen. Wir wollen aber den dritten Assembler-Befehl kennenlernen:

RTS

Das heißt »return from subroutine«, also »Rückkehr aus Unterprogramm«. In unserem Fall bewirkt das eine Rückkehr zum Basic. Wie Sie sehen, ist das ein 1-Byte-Befehl, also ohne Argument. Auch hier spricht man von einer Adressierungsart, nämlich der »impliziten« Adressierung. Man erkennt sie am Fehlen des Argumentes. Die Adresse ist implizit, das heißt im Befehl selbst enthalten. Dies ist nämlich ein Befehl, der immer an den Programmzähler gerichtet ist. Der Computer holt sich vom Stapel-Speicher die dort zuoberst liegende Adresse, das ist die, bei der der Computer in ein Unterprogramm gesprungen ist oder aber die, bei der der Computer Basic verlassen hat. Wir ergänzen also noch:

A 150A RTS

PC	JRQ	NV-BDJZC	AC	XR	YR	SP
E147	EA31	10110000	00	00	00	F8

Bild 1. Eine Registeranzeige

N	V	—	B	D	J	Z	C
Negativ-Flagge	Überlauf-Flagge	unbenutzt	Abbruch-Flagge	Dezimal-Flagge	Interrupt-Flagge	Zero-(Null) Flagge	Carry-(Übertrag) Flagge

Bild 2. Das Prozessor-Status-Register P: die Flaggen

Befehlswort	Adressierung	Byteanzahl	Code HEX	Code DEZ	Dauer in Taktzyklen	Beeinflussung von Flaggen
LDA	unmittelbar	2	A9	169	2	N, Z
	absolut	3	AD	173	4	N, Z
LDX	unmittelbar	2	A2	162	2	N, Z
	absolut	3	AE	174	4	N, Z
LDY	unmittelbar	2	A0	160	2	N, Z
	absolut	3	AC	172	4	N, Z
STA	absolut	3	8D	141	4	keine
STX	absolut	3	8E	142	4	keine
STY	absolut	3	8C	140	4	keine
RTS	implizit	1	60	96	6	keine

Bild 3. Die ersten sieben Befehle

und starten das Programm, zum Beispiel von Basic aus mit SYS 5376. Natürlich taucht dann in der linken oberen Ecke des Bildschirmes ein schwarzes A auf. Hier noch der Basic-Lader:

```
10 FOR I=5376 TO 5386:READ
A:POKE I,A:NEXT I:END
20 DATA 169,1,141,0,4*,169,0,141,0,216*,
96.
```

Die mit * markierten Zahlen müssen für den VC 20 verändert werden: Grundversion: 30 und 148 Erweiterung: 16 und 150.

Eine Kombination von LDA mit STA ist vergleichbar mit dem POKE-Befehl in Basic. Man kann in Assembler nicht direkt eine Zahl in einen Speicher einschreiben, sondern muß den Umweg über den Akku machen. Außer dem Akku eignen sich dazu aber auch das X-Register und das Y-Register. Hierfür gibt es die Befehle LDX (lade X-Register), STX (lege X-Register-Inhalt ab), LDY lade Y-Register) und schließlich STY (lege Y-Register-Inhalt ab). Sie können das übungshalber an unseren kleinen Programm ausprobieren. An dem folgenden Programm sehen Sie noch eine Eigenart der drei Register (Akku, X-Register, Y-Register):

```
A 1500 LDA #$01
A 1502 LDX #$00
A 1504 LDY #$02
A 1506 STA $0400
A 1509 STX $D800
A 150C STY $0401
A 150F STX $D801
A 1512 STA $0402
A 1515 STX $D802
A 1518 RTS
```

Für den VC 20 werden die entsprechenden Speicherstellen für Bildschirm- und Bildschirmfarbspeicher eingesetzt. Dieses Programm druckt — wie erwartet — »ABA« in die linke obere Ecke des Bildschirms. Dabei ist das X-Register dreimal ausgelesen worden und der Akku zweimal. Sie sehen also, daß die Registerinhalte durch die STA-, STX-, STY-Befehle nicht verändert werden.

Wir wollen noch etwas ausprobieren. Bisher haben wir den LDA-Befehl nur mit der »unmittelbaren« Adressierung kennengelernt. LDA, LDX, LDY können auch »absolut« adressiert werden.

```
A 1518 LDA $D800
```

Damit laden wir den Inhalt der Speicherstelle $ D800 (beim VC 20 die anderen Adressen des Bildschirmfarbspeichers) in den Akku. Der Inhalt ist seit $1509 eine Null. Jetzt weiter:

```
A 151B STA $0403
A 151E STX $D803
A 1521 RTS
```

Das müßte beim Ablauf des Programms noch einen Klammeraffen (@mit Bildschirmcode 0) an die vierte Stelle plazieren, was Sie durch SYS 5376 leicht nachprüfen können. Sie sehen, daß man mit diesen sieben Befehlen schon eine Menge anfangen kann.

Wir kommen noch einmal zur Adressierung. Ich hatte Ihnen gesagt, daß LDA #$01 ein 2-Byte-Befehl mit unmittelbarer Adressierung ist (ein Byte für LDA und eines für 01), LDA $D800 ist ein 3-Byte-Befehl (ein Byte für LDA, je eines für das LSB und das MSB von $D800) mit absoluter Adressierung. Da werden Sie sich doch sicher schon gefragt haben, wo bleibt die Adressierung! Wenn aber kein Byte für die Adressenmarkierung (zum Beispiel #) reserviert ist, muß die Kennzeichnung irgendwie anders sein. Wenn Sie einen Disassembler zur Verfügung haben, dann sehen Sie sich damit unser Programm an. Fast jeder Disassembler gibt neben dem Assemblertext auch Byte für Byte in Hexadezimalzahlen die Codes an. Wenn Sie nun die beiden Befehle LDA #$01 und LDA $d800 von den Codes her untersuchen, sehen Sie folgendes:

```
1500 A9 01        LDA #$01
und
1518 AD 00 D8     LDA $D800
```

Offensichtlich gehört jeweils das erste angezeigte Byte zu LDA. Sie sind aber verschieden! Wir sehen daraus, daß die Codezahl für einen Befehl gleich zwei Informationen enthält: Das Befehlswort selbst (LDA) und die Adressierungsart.

Genauso wie man LDA sowohl unmittelbar als auch absolut ausführen kann, ist das auch mit LDX und LDY möglich. Bei den Befehlen STA, STX, STY ist eine unmittelbare Adressierung sinnlos. Für RTS kennt man nur eine implizite Adressierung. Wir fassen das alles zusammen in Bild 3.

In den letzten Spalten von Bild 3 ist noch angegeben, inwieweit durch diese Befehle das Prozessorstatusregister beeinflußt wird, so wie wir es für den Befehl LDA schon ausprobiert haben. In der vorletzten Spalte sehen Sie, wie lange die Ausführung eines Befehls dauert. Wenn sie für einen Taktzyklus etwa eine Mikrosekunde rechnen, dann müßten Sie jetzt ausrechnen können, wie lange unser letztes Programm zur Bearbeitung braucht: 48 Mikrosekunden. Ein vergleichbares Basic-Programm braucht dazu etwa hundertmal so lange: zirka 0,05 Sekunden.

Ein bißchen von Assembler-Alchimie verstehen Sie jetzt schon mit diesen sieben Befehlen. Wir wollen uns nun die Zahlen ansehen, die hier Verwendung finden: Das Binärsystem und das Hexadezimalsystem.

Die einzigen Ziffern, die unser Computer kennt, sind 0 und 1. Sie stehen für »Strom an« oder »Strom aus«, oder für »keine magnetische Erregung« oder »magnetische Erregung«. Deswegen ist es für uns als angehende Assembler-Alchimisten von großer Bedeutung — wir arbeiten ja ganz eng an der Hardware — dieses binäre Zahlensystem handhaben zu können. Das Hexadezimalsystem kennt der Computer eigentlich gar nicht. Wir verwenden es, weil es in einem besonders engen Zusammenhang mit Binärzahlen und dem Aufbau unseres Computers steht: Die größte einstellige Hex-Zahl ist $F, das entspricht genau 1111 im Binärsystem, also dem maximalen Füllungsgrad eines halben Bytes, das Nibble genannt wird. Ein ganzes Byte kann maximal $FF enthalten (binär 1111 1111) und der gesamte Speicheradressenbereich unseres Computers geht bis $FFFF (dezimal 65535). Eine einstellige Hex-Zahl paßt also in ein Nibble, eine zweistellige in ein Byte und eine dreistellige oder vierstellige in zwei Bytes, weshalb man solche Hex-Adressen auch recht leicht in das LSB und das MSB aufteilen kann:

```
$ D8  00
  MSB LSB
```

Rechnen werden wir mit Hexadezimalzahlen nicht, dazu benutzen wir dann das Dezimalsystem oder — wenn es sich um computerinterne Vorgänge handelt — das Binärsystem.

Das Rechnen mit Binärzahlen funktioniert genauso wie das mit Dezimalzahlen. Es gilt also

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=10$

wobei binär 10 gleich dezimal 2 ist. Als Beispiel können wir mal 2+1=3 im Binärsystem rechnen:

```
 10 entspricht dez. 2
+01 entspricht dez. 1
---
 11, was ja dezimal 3 ergibt.
```

Die Addition erfolgt also spaltenweise wie beim gewohnten dezima-

Fortsetzung Seite 179

In die Geheimnisse der Floppy eingetaucht

Das Diskettenlaufwerk VC 1541 ist der Renner unter den Massenspeichern. Doch mit der passenden Literatur hapert es. Deshalb beschränken sich die meisten Anwender auf das Speichern und Laden von Programmen. Mit diesem Kurs lernen Sie, Ihre Floppy effektiv auszunützen und schließlich zu manipulieren.

Daß die 1541 ein sehr wandelbares Gerät ist, werden die meisten Benutzer wohl wissen oder zumindest erahnen. Man denke ja nur an den »Kleinkrieg« zwischen Softwareherstellern und Softwarepiraten, die sich gegenseitig das Leben schwer machen. Die meisten »Schlachten« liefert man sich hier im Inneren der Floppy, die viel raffiniertere Methoden des Programmschutzes anbietet als der Commodore 64.

Aber auch solche Programme, wie das HYPRA-LOAD, das Sie ebenfalls in dieser Ausgabe als Listing des Monats finden, beweisen die Flexibilität der 1541. Doch wie bei so vielen Dingen in der Commodore-Welt sind auch hier die Informationen rar, beziehungsweise in den Handbüchern gar nicht vorhanden. So wollen wir uns mit Ihnen an die Floppy heran- und in sie hineintasten. Angefangen bei grundlegenden Informationen über den Diskettenaufbau und den Befehlssatz der Floppy werden wir Schritt für Schritt in deren Möglichkeiten zur Programmierung und Manipulation hinabtauchen. Was wird benötigt?

Nun, außer einem C 64 und einer VC 1541, »nur« Basic-Erfahrungen, grundlegende Kenntnisse in Maschinensprache (für spätere Folgen) und ein wenig Geduld.

Bevor wir jedoch mit unserer ersten Tauchfahrt beginnen, tippen Sie bitte das beigefügte Programm EDDI (Listing 1) ein, sofern Sie nicht über einen eigenen Disk-Monitor verfügen. Auf die Bedienung von EDDI wird im Einzelnen eingegangen.

Sehen wir uns jetzt erst einmal so eine Diskette an; die folgenden Erläuterungen beziehen sich auf eine formatierte Diskette.

Aufbau einer Diskette

Die Diskette ist in 35 konzentrische Spuren (englisch: Tracks) aufgeteilt. Diese Spuren enthalten wiederum jede eine bestimmte Anzahl von Sektoren, die von außen nach innen abnimmt. Die genauen Zahlenverhältnisse stehen in Tabelle 1.

Die Spuren sind, beginnend mit der äußeren Spur, von 1 bis 35 durchnumeriert. Die Sektoren sind auf den Spuren in numerischer Reihenfolge gegen den Uhrzeigersinn angeordnet. Jeder Sektor enthält einen Block, das sind 256 Bytes, an Information. Es kann jeder der 683 Blöcke auf der Diskette durch Angabe der jeweiligen Spur- und Sektornummer aufgerufen werden. Allerdings stehen davon dem Benutzer normalerweise nur 664 Blocks zur Verfügung, da das Betriebssystem der Floppy die Spur 18 für sich beschlagnahmt.

Für die nun folgenden Versuche wäre es sinnvoll, eine Diskette neu zu formatieren, mit der wir ein bißchen »spielen« können. Sehen wir uns nun erst einmal das Directory an (LOAD "$",8).

In der ersten Zeile stehen die Drivenummer (hier immer 0) und der Name der Diskette, sowie die ID und das Formatkennzeichen (genaueres später).

Die zweite Zeile enthält, da sich kein File auf der Diskette befindet, die Meldung »664 BLOCKS FREE«.

Erste Versuche mit EDDI, dem Disk-Monitor/Editor

Da sich diese Informationen auf der schon erwähnten Spur 18 befinden, wollen wir uns diese Spur mit EDDI gleich einmal etwas genauer ansehen. Laden Sie den Editor und legen Sie unsere »Spieldiskette« ein; danach starten Sie mit RUN.

Als Kommando tippen Sie F3 für »BLOCK LESEN«. Danach geben Sie, durch Komma getrennt, die Spur und Sektornummer des gewünschten Blocks ein; in unserem Fall »18,0«.

Nach dem Ladevorgang meldet sich EDDI mit Byte 0 der ersten von 16 Seiten, zu je 16 Bytes. Drücken Sie jetzt RETURN, um die erste Seite anzuzeigen, welche wir nun betrachten wollen.

Es sollte vielleicht erwähnt werden, daß die Zählung von Blöcken und Bytes grundsätzlich bei Null beginnt. Den eingeladenen Block bezeichnet man als BAM (Block Availability Map), auf deutsch etwa »Blockbelegungsplan«. Dieser Plan gibt an, welche Blöcke auf der Diskette frei und welche schon beschrieben sind. Ferner enthält er den Namen der Diskette, die ID, das Formatkennzeichen und den Beginn des Directory.

Die ersten beiden Bytes (0,1) dieses Blocks enthalten Spur und Sektor des ersten Directoryblocks; normalerweise »18,1« (siehe auch Tabelle 2).

Byte 2 enthält das Formatkennzeichen (hier 65, beziehungsweise »A«). Zur Erklärung: Commodore stellt ja verschiedene Laufwerke her, zum Beispiel 1541, 3040, 8050, 8250... Diese Laufwerke unterscheiden sich fast alle im Aufzeichnungsformat, das heißt Anzahl und Verteilung der Spuren und Sektoren; so hat die CBM 8050 77 Spuren mit bis zu 29 Sektoren, was deren höhere Speicherkapazität zur Folge hat. Solche Disketten können verständlicherweise von der 1541 weder gelesen noch beschrieben werden. Am Formatkennzeichen »A« erkennt die 1541 nun Disketten ihres eigenen Formats; ist dieses nicht identisch, so beschwert sich die Floppy mit einer Fehlermeldung. Eine Ausnahme dieser Regel bildet die Lesekompatibilität, die besagt, daß eine »fremde« Diskette zwar gelesen, aber nicht beschrieben werden kann (zum Beispiel 3040 auf 1541).

Byte 3 steht generell auf Null, da es bei der 1541 keine Funktion erfüllt.

In die Geheimnisse der Floppy eingetaucht

Die Bytes 4 bis 143 enthalten nun die eigentliche BAM, deren Format ein wenig kompliziert ist: Für jede Spur sind 4 Bytes reserviert, wobei das jeweils erste Byte die Anzahl der noch freien Blöcke auf dieser Spur angibt. Die folgenden drei Bytes müssen wir als eine Gesamtheit von 24 Bits betrachten, wobei jedes gesetzte Bit einen freien Block signalisiert; siehe auch Tabelle 3.

Um auch die folgenden Seiten des Blocks zu betrachten, drücken Sie zum Vorwärtsblättern F1; die weitere Bedienung ist analog zur oben beschriebenen. Rückwärtsblättern ist durch Drücken von F2 möglich.

Fahren Sie nun bis zum Byte 144 vor und sehen Sie sich die Seite an.

Die Bytes 144 bis 161 enthalten den Namen der Diskette, der beim Formatieren festgelegt wird. Direkt im Anschluß daran folgen die Bytes 162,163, die die ID im ASCII-Code beinhalten, gefolgt von einem »Shift Space«. An der ID erkennt die Floppy, ob die Diskette gewechselt wurde; deshalb sollte jede Diskette eine andere ID haben.

Byte 165 und 166 enthalten DOS-Version und Formatkennzeichen, hier normalerweise »2A«, wiederum gefolgt von einem »Shift Space«.

Die Bytes 171 bis 255 haben normalerweise keine Bedeutung und können unterschiedlich gefüllt sein.

Wie sieht das Inhaltsverzeichnis aus?

Auf unserer Entdeckungsreise durch Spur 18 folgen wir jetzt der Angabe in den ersten beiden Bytes und laden den ersten Directoryblock (F3; 18,1). Das Format des Blocks ist der Tabelle 4 zu entnehmen. Jeder Direktoryblock enthält acht File-Einträge und den Zeiger auf den nächsten Direktoryblock (Byte 0 und 1); ist die Tracknummer des nächsten Blocks 0, so war der gelesene Directoryblock der letzte, und das zweite Byte zeigt die Anzahl der hier benutzten Bytes an. In unserem Fall stehen hier 0 und 255.

Nun zu Tabelle 5, die das Format eines Directoryeintrags darlegt: Jeder dieser Einträge besteht aus 30 Bytes, wobei das erste den Filetyp (siehe Tabelle 6), die beiden nächsten Spuren und Sektoren des ersten Fileblocks und die 16 folgenden Bytes den Filenamen enthalten. Die folgenden 3 Byte werden nur bei relativen Dateien verwendet; sie werden später im einzelnen noch besprochen.

Byte 26 und 27 enthalten Track und Sektor des neuen Files, falls das alte mit »@« überschrieben wurde. Die Bytes 28,29 schließlich geben die Anzahl der belegten Blöcke dieses Files an.

Die einzelnen Datei-Typen der Floppy

Diese bis jetzt beschriebenen Angaben werden vom Betriebssystem der Floppy, also vom DOS (englisch: Disk Operating System) verwaltet.

Beschäftigen wir uns nun mit den restlichen Blöcken auf der Diskette, die dem Anwender zur freien Verfügung stehen, denn dort werden die einzelnen Files abgespeichert, deren Aufbau uns jetzt interessiert.

DEL-Files:

Diese Fileanzeige existiert normalerweise nicht im Directory; wird ein File gelöscht, so wird dieses nicht mehr angezeigt; das Byte des Filetyps steht dann auf 0. Durch setzen des Filetyps auf 128 (hex. $80) kann eine DEL-Anzeige jedoch erzwungen werden.

SEQ-Files:

Dieser Filetyp dient zur Speicherung von Daten auf Diskette (im Gegensatz zur Programmspeicherung). Der Aufbau dieses Filetyps ist relativ einfach: Die ersten beiden Bytes eines Datenblocks zeigen jeweils auf den nächsten Block im File; so erfolgt eine beliebig lange Blockverkettung auf der Diskette. Da aber auch das schönste File einmal zu Ende geht, muß der letzte Block gekennzeichnet sein. Dies erfolgt, wie schon beim Directory, durch eine 0 als Spurnummer. Die Sektornummer bezeichnet jetzt die Anzahl der belegten Datenbytes dieses Blocks. Diese Art der Verkettung von Blöcken wird bei allen Filetypen vorgenommen! Die restlichen 254 Bytes jedes Blocks enthalten die Daten.

USR-Files

USR-Files stimmen im Aufbau exakt mit den SEQ-Files überein, sie haben jedoch noch Zusatzfunktionen im DOS, auf die ein anderes Mal eingegangen werden soll.

PRG-Files

PRG-Files stellen den häufigsten Filetyp dar. Sie dienen der Speicherung von Programmen auf der Diskette und haben nahezu den selben Aufbau wie SEQ-Files. Der einzige Unterschied besteht in den Bytes 2 und 3 des ersten Blocks, welche die Startadresse des Programms im Computer enthalten. Ist diese Adresse gleich der Adresse des Basic-Anfangs, also 2049 ($0801), so können die Programme mit »LOAD"Name",8« geladen werden; dieser Modus ignoriert die Anfangsadresse auf Diskette und lädt die Programme generell an den Basic-Anfang (sogenanntes relatives Laden). Sollen Programme jedoch an anderen Stellen im Speicher stehen, zum Beispiel Maschinenprogramme, so muß diese angegebene Adresse als Startadresse benutzt werden; man lädt hier absolut mit »LOAD"Name",8,1«.

REL-Files:

Dieser Filetyp ist im Aufbau ungleich komplizierter als die eben besprochenen; es soll daher zuerst kurz auf die Arbeitsweise von REL-Files eingegangen werden. Sequentielle Files haben den Nachteil, daß sie praktisch nur aus einem Datensatz bestehen. Sucht man nun, zum Beispiel in einer Kartei, eine bestimmte Hausnummer oder einen bestimmten Namen, so muß der gesamte Datensatz durchgelesen werden, um die entsprechende Stelle zu finden. In einer relativen Datei geht man deshalb einen anderen Weg, um jede Stelle schnell auffinden zu können.

Es existiert eine beliebige Anzahl (zum Beispiel 100) von Datensätzen, wobei alle Datensätze die gleiche Länge haben müssen (maximal 254 Zeichen).

Das DOS legt jetzt einen sogenannten Side-Sektor an, der aus bis zu sechs Blöcken bestehen kann. Diese Blöcke enthalten nun die Zeiger auf sämtliche Datenblöcke, in

denen die Datensätze gespeichert sind (1 Datensatz hat maximal 1 Block Länge). Auch hier sind die Datenblöcke wieder durch Zeiger in den Bytes 0 und 1 verkettet. Den Aufbau eines Side-Sektor-Blocks zeigt Tabelle 7. Zum besseren Verständnis hier ein kleines Beispiel:

```
10 REMEDDI - DISKMONITOR/EDITOR
20 REM          VON
30 REMKARSTEN SCHRAMM      1984
40 :
50 PRINT"□■":POKE53280,14:POKE53281,14
60 GOSUB10000
70 OPEN1,8,15,"I0":OPEN2,8,2,"#"
80 PRINT"□ E D D I  -  HAUPTMENUE"
85 HE$="BYTE        DEC   HEX   BIN
 ASC":POKE650,128
90 PRINT"────────────────────────"
100 PRINT:PRINT:PRINT
110 PRINT"(F1) - SCROLLING VORWAERTS":PR
INT
120 PRINT"(F2) - SCROLLING RUECKWAERTS":
PRINT
130 PRINT"(F3) - BLOCK LESEN":PRINT
140 PRINT"(F4) - BLOCK SCHREIBEN":PRINT
150 PRINT"(F5) - EDITOR EINSCHALTEN":PRI
NT
160 PRINT"(F6) - DISKETTE WECHSELN":PRIN
T
170 PRINT"(F7) - RUECKKEHR INS MENUE":PR
INT
180 PRINT"(F8) - PROGRAMMENDE"
190 PO=1:GOTO 9000
1000 REM EDDI AN
1010 X=0:Y=0
1020 FORY=ETO255STEP16
1030 PO=2:PRINT"□EDITOR-MODUS FUER TRACK
"T" SEKTOR"S
1040 PRINT:PRINTHE$:PRINT
1050 FORX=YTOY+15:PRINTX:NEXTX
1060 PRINT"■■■■":FORX=YTOY+15
1065 DA=PEEK(50000+X):GOSUB7030:PRINTX,O
U$
1070 INPUT"□■■■■■■■■";IN$:IFIN$=""THEN10
90
1072 IFLEFT$(IN$,1)="↑"THENPRINT"■■■■■■■
":GOTO9000
1073 IFLEFT$(IN$,1)="←"THENPRINT"■■■■■■■
":GOTO1125
1075 DA=VAL(LEFT$(IN$,3)):IFDA>255ORDA<0
THENPRINT"□□":GOTO1065
1080 POKE50000+X,DA
1120 NEXTX:PRINT
1125 PRINT"EINGABE ?";
1130 GETA$:IFA$=""THEN1130
1140 IFA$="■"THEN1200
1150 IFA$="■"THEN1300
1160 IFA$<>" "THENNEXTY
1170 PO=1:GOTO9000
1200 PRINT"■■■■":PRINTE".....???"
1210 GETA$:IFA$=""THEN1210
1215 IFA$="■"THEN1300
1220 IFA$<>"■"THEN1020
1230 E=E+16:IFE>255THENE=0
1240 GOTO1200
1300 PRINT"■■■■":PRINTE".....???"
1310 GETA$:IFA$=""THEN1310
1315 IFA$="■"THEN1200
1320 IFA$<>"■"THEN1020
1330 E=E-16:IFE<0THENE=240
1340 GOTO1300
2000 REM DISKETTENWECHSEL
2010 PRINT"□BITTE NEUE DISKETTE EINLEGEN
"
2020 GETA$:IFA$=""THEN2020
2030 RUN
3000 REM BLOCK READ
3010 PO=2:PRINT"□   BLOCK LESEN":PRINT:P
RINT
3020 INPUT"TRACK, SEKTOR ";T,S
3025 IFT<1ORT>35THEN3010
3030 PRINT#1,"U1 2 0"T;S
3035 IFST<>0THENPRINT:GOTO9000
3040 PRINT#1,"B-P 2 0"
3050 SYS49152:E=0:X=0:Y=0:GOTO5010
3060 FORY=ETO255STEP16
3070 PRINT"□TRACK"T" SEKTOR"S
3080 PRINT:PRINTHE$:PRINT
3090 FORX=YTOY+15:DA=PEEK(50000+X):GOSUB
7030:PRINTX,OU$:NEXTX
3100 GOTO9000
4000 REM BLOCK WRITE
4010 PO=1:PRINT:PRINT:INPUT"□■TRACK, SEK
TOR";T,S:PRINT"■"
4020 PRINT#1,"B-P 2 0"
4030 SYS49177
4040 PRINT#1,"U2 2 0"T;S
4050 GOTO9000
5000 REM SCROLL FORWARD
5010 E=X:IFE>255THENX=0:E=0
5020 PRINT"□TRACK"T" SEKTOR"S
5030 PRINT:PRINTHE$:PRINT
5040 DA=PEEK(50000+E):GOSUB7030:PRINTE,O
U$
5050 X=X+16
5060 GETA$:IFA$=""THEN5060
5070 IFA$="■"THEN5010
5075 IFA$="■"THENX=X-16:GOTO6010
5077 IFA$="■"THEN1000
5080 GOTO3060
6000 REM SCROLL BACKWARD
6010 E=X:IFE<0THENE=240:X=240
6020 PRINT"□TRACK"T" SEKTOR"S
6030 PRINT:PRINTHE$:PRINT
6040 DA=PEEK(50000+E):GOSUB7030:PRINTE,O
U$
6050 X=X-16
6060 GETA$:IFA$=""THEN6060
6070 IFA$="■"THEN6010
6075 IFA$="■"THENX=X+16:GOTO5010
6077 IFA$="■"THEN1000
6080 GOTO3060
7000 REM BEREITSTELLUNG DES STRINGS
7010 REM DA/DA$ SIND AUSGABEWERTE
         H$,D$,B$,C$ SIND ZWISCHENWERTE
7020 REM OU,OU$ SIND ENDERGEBNISSE
7030 IFDA>31ANDDA<128ORDA>159ANDDA<256TH
ENC$=CHR$(DA):GOTO7040
7035 C$="."
7040 XX$="000":D$=RIGHT$(STR$(DA),LEN(ST
R$(DA))-1)
7045 D$=LEFT$(XX$,3-LEN(D$))+D$
7050 XX$="123456789ABCDEF":H$=""
7060 HH=INT(DA/16):HL=DA-HH*16
7070 IFHHTHENH$=H$+MID$(XX$,HH,1):GOTO70
80
7075 H$=H$+"0"
7080 IFHLTHENH$=H$+MID$(XX$,HL,1):GOTO70
90
7085 H$=H$+"0"
7090 B$="":FORQ=7TO0STEP-1
7100 IF(DAAND(2↑Q))<>0THENB$=B$+"1":NEXT
:GOTO7110
7105 B$=B$+"0":NEXT
7110 OU$=D$+"  "+H$+"   "+B$+"  "+C$
7120 RETURN
8999 END
9000 REM GET KOMMANDO
9010 PRINT:PRINT"KOMMANDO ?  ";
9020 PRINT"■■";:FORW=1TO75:GETKO$:IFKO$<
>""THEN9090
9030 NEXTW
9040 PRINT"■■■■";:FORW=1TO75:GETKO$:IFKO
$<>""THEN9090
9050 NEXTW
9060 GOTO9020
9090 IFKO$="@"THEN9200
9100 IF ASC(KO$)>140ORASC(KO$)<133THEN90
20
9110 KO=ASC(KO$)-132
9120 ON PO GOTO9130,9140,20000
9130 ON KO GOTO9020,3000,1000,80,9020,40
00,2000,20000
9140 ON KO GOTO5000,3000,1000,80,6000,40
00,2000,20000
9200 PRINT
9210 GET#1,A$:PRINTA$;:IFST<>64THEN9210
9220 GOTO 9000
9999 END
10000 DATA160,0,169,8,32,9,237,169,98,32
,199,237,32,19,238,153,80,195,200
10010 DATA208,247,32,239,237,96,160,0,16
9,8,32,12,237,169,98,32,185,237
10020 DATA185,80,195,32,221,237,200,208,
247,32,254,237,96,0,0
10030 RESTORE:FORZ=1TO51:READA:POKE49151
+Z,A:NEXT
10040 REM GET:49152; WRITE:49177
10050 RETURN
20000 PRINT:PRINT:PRINT"□AUF WIEDERSEHEN
 !!!":PRINT:POKE53280,14:POKE53281,6
20001 PRINT"UND DANKESCHOEN !"

READY.
```

Listing 1. EDDI, ein Disk-Monitor/Editor

Wir haben eine relative Datei mit 250 Datensätzen à 127 Zeichen. Diese Datei benötigt also 125 Datenblöcke und zwei Side-Sektor-Blöcke. Im Directory-Eintrag finden wir jetzt die schon erwähnten zusätzlichen Bytebelegungen: Byte 19 und 20 jedes Eintrags enthalten jetzt Spur und Sektor des ersten Side-Sektor-Blocks; Byte 21 gibt die Datensatzlänge (Recordlänge) an.

Wir wollen jetzt auf den 248. Datensatz zugreifen; das DOS arbeitet nun folgendermaßen: Ein Datensatz enthält 127 Byte, das heißt, es passen zwei Datensätze in einen Block; dadurch errechnet sich der Block, auf den jetzt zugegriffen wird, aus (248-1)/2=123.5. (Minus 1, da immer von 0 an gezählt wird). Da ein Side-Sektor-Block nur 120 Einträge aufnehmen kann, ist der Zeiger auf den Datenblock im Side-Sektor-Block Nummer 2 zu finden. Dieser wird jetzt anhand des Verzeichnisses in Block 1 gelesen und dann auf Zeiger Nummer 3 (Bytes 22,23) zugegriffen. Wir kennen also jetzt Spur und Sektor des Blockes, in dem unser Datensatz steht; die Position des ersten Datenbyte berechnet sich jetzt aus dem Nachkommaanteil der obigen Division (0.5*254=127). Der Datensatz beginnt also beim 127+2=129ten Byte.

Der Aufbau von relativen Dateien ist also, wie schon erwähnt, ziemlich kompliziert; diese Art der Datenspeicherung hat aber einige Vorteile gegenüber der 'normalen' mit SEQ-Files.

Bedienungshinweise für EDDI, dem Disk-Monitor/Editor

Da unserem U-Boot auf dieser schwierigen Fahrt der Sauerstoff ausgegangen ist, wollen wir uns nun erst einmal erholen. Hier noch ein paar Anregungen zur Arbeit mit EDDI: EDDI kann nicht nur Blöcke lesen und anzeigen; Sie können auch Bytes verändern und diesen Block danach wieder abspeichern.

Dazu laden Sie den zu verändernden Block und fahren auf die Seite, die Sie interessiert; hier tippen Sie als Kommando F5, und der Editor-Modus startet. Sie können jetzt Bytes dezimal abändern, indem Sie den jeweils neuen Wert eingeben und »RETURN« drücken. Wollen Sie aus dem Eingabemodus aussteigen, so

In die Geheimnisse der Floppy eingetaucht

Spuren	Sektoren	Anzahl Total
01-17	00-20	21
18-24	00-18	19
25-30	00-17	18
31-35	00-16	17

Tabelle 1. Spuren und Sektoren des VC 1541-Diskettenformates

Aufbau des Blocks 18,0:

BYTE(s)	Bedeutung:
000-001	Spur und Sektor des ersten Directory-Blocks
002	Formatkennzeichen »A«
003	unbenutzt
004-007	BAM der Spur 1
008-011	BAM der Spur 2
012-143	BAM der Spuren 3-35
144-161	Name der Diskette, aufgefüllt mit 160
162-163	ID der Diskette
164	unbenutzt
165-166	DOS und Formatkennzeichen
167-255	nicht benutzt

Tabelle 2. Aufbau und Inhalt der BAM (Block-Belegungs-Plan) in Spur 18, Sektor 0

Aufbau eines 4 Byte-Eintrages in der BAM (eine Spur)

BYTE(s)	Bedeutung:
000	Anzahl der freien Blöcke dieser Spur
001-003	Belegplan der Spur Jedes Byte ist zuständig für 8 Sektoren: Byte 1 für 0-7 Bit 7 für Sektor 0 Bit 6 für Sektor 1 und so weiter Byte 2 für 8-15 Byte 3 für 16-23

Tabelle 3. Für jede Spur reserviert die BAM 4 Bytes

Aufbau eines Directory-Blocks:

BYTE(s)	Bedeutung
000-001	Spur und Sektor des nächsten Dir.-Blocks
002-031	Eintrag Nr. 1
032-033	unbenutzt
034-063	Eintrag Nr. 2
064-066	unbenutzt
067-225	Einträge Nr. 3-7 bzw. unbenutzt
226-255	Eintrag Nr. 8

Tabelle 4. Aufbau der Directory der VC 1541

tippen Sie entweder »RETURN« und können weiterblättern ohne den Editor zu verlassen, oder Sie tippen »↑ RETURN«, um in den Kommando-Modus zu kommen. Nach einigem Probieren wird Ihnen EDDI sehr schnell vertraut werden; wir gehen auch in den folgenden Ausgaben noch darauf ein.

Wichtig:

Beim Wechseln einer Diskette muß die Funktionstaste F6 getippt und nach dem Austausch eine Taste gedrückt werden, sonst reagiert die Floppy mit einer Fehlermeldung. Diese können übrigens mit »@« abgerufen werden. Das Zurückschreiben eines Blocks auf Diskette erfolgt mit F4, wobei Spur und Sektornummer angegeben werden müssen. Hier noch ein paar Vorschläge zum Ausprobieren: Ändern Sie doch einmal auf Ihrer Versuchsdiskette (!) das Formatkennzeichen (Spur 18, Sektor 0, Byte 2 auf 66 statt jetzt 65 und speichern den Block an die gleiche Stelle auf die Diskette zurück. Versuchen Sie nun einmal ein kleines Programm auf diese Diskette zu schreiben. (Die genauen Vorgänge in der Floppy werden beim nächstenmal erläutert.) Oder ändern Sie einmal die Bytes im Directory, die den Filetyp angeben, entsprechend Tabelle 6 und laden Sie es danach. Mit dem klugen Satz »Probieren geht über Studieren« verabschieden wir uns für diese Ausgabe. Nächstesmal beginnt dann ein praktischer Teil unserer Expedition, nämlich die Vorstellung des Befehlssatzes der 1541 mit vielen Beispielen und Anregungen.

(K. Schramm/B. Schneider/gk)

Aufbau des Filetyp-Bytes

BIT	Bedeutung, in Klammern jeweiliger Inhalt				
0	(0)	(1)	(0)	(1)	(0)
1	(0)=DEL	(0)=SEQ	(1)=PRG	(1)=USR	(0)=REL
2	(0)	(0)	(0)	(0)	(1)
3	unbenutzt				
4	unbenutzt				
5	unbenutzt				
6	(0)=normal; (1) = File kann durch SCRATCH nicht mehr gelöscht werden				
7	(0)= File noch offen (1)= File ordnungsgemäß geschlossen				

Tabelle 6. Die Bedeutung des ersten Bytes eines Directory-Eintrages

Aufbau eines Directory-Eintrags:

BYTE(s)	Bedeutung
000	Filetyp, siehe ges. Tab
001-002	Spur und Sektor des ersten Datenblocks
003-018	Filename, aufgefüllt mit 160
019-020	REL-Files: Spur und Sektor des ersten Side-Sektor-Blocks
021	REL-Files: Datensatzlänge
022-025	unbenutzt
026-027	Spur und Sektor beim Überschreiben mit @ (nur Zwischenspeicher)
028-029	Anzahl der von diesem File belegten Blocks

Tabelle 5. Bedeutung der einzelnen Byte des Directory

Aufbau eines Side-Sektor-Blocks:

BYTE(s)	Bedeutung
000-001	Spur und Sektor des nächsten Side-Sektor-Blocks
002	Nummer des Side-Sektor-Blocks
003	Datensatzlänge
004-005	Spur und Sektor des Side-Sektor-Blocks 1
006-007	Spur und Sektor des Side-Sektor-Blocks 2
008-015	Spur und Sektor der Side-Sektor-Blöcke 3-6
016-017	Spur und Sektor des ersten Datenblocks für den der Side-Sektor-Block zuständig ist. (Datenblock 0) Spur und Sektor des zweiten Datenblocks (Nr. 1)
018-255	Spur und Sektor der Datenblocks Nr. 2 bis Nr. 119

Tabelle 7. Relative Dateien benutzen Side-Sektor-Blöcke um Datensätze gezielt anzuspringen

Der gläserne VC 20

In der ersten Folge haben wir uns hauptsächlich mit der Basic-Verwaltung befaßt. Dabei tauchte immer wieder der Begriff Zeropage auf, mit dem wir uns heute beschäftigen wollen.



Die Zeropage — oder zu Deutsch die Seite Null — ist in Maschinensprache besonders einfach zu handhaben. Als Seite bezeichnet man im übrigen immer ein Paket von jeweils 256 Byte. So entspricht Seite 0 den Adressen 0-255, Seite 1 den Adressen 256-511 und so weiter.

Eine Zeropageadressierung, zum Beispiel LDA $42, benötigt nur zwei Byte, eine absolute Adressierung, zum Beispiel LDA $1234, hingegen drei Bytes im Speicher. Damit verbunden ist auch die Bearbeitungsgeschwindigkeit eines Maschinenprogramms. Denn die Zeropageadressierung ist schneller als die entsprechende Drei-Byte-Methode. Bei kleineren Programmen in Assembler fällt dieser Aspekt zwar nicht so sehr ins Gewicht, bei sehr umfangreichen Routinen, (wie zum Beispiel beim Basic-Interpreter) spielt die Adressierungsart jedoch eine größere Rolle.

In der Seite 0 legt der Computer also insbesondere die Daten ab, die er oft benötigt, wie zum Beispiel Vektoren, Parameter etc. Die komplette Liste der Adreßbelegung zeigt Tabelle 1.

Wir wollen es jedoch nicht nur mit der Aufstellung alleine bewenden lassen. Die interessantesten Lokationen möchte ich hier herausgreifen und besprechen. Beginnen wir also mit den ersten drei Bytes im Speicher:

Der USR-Vektor

Adresse 0,1,2: Über die USR-Funktion kann der Benutzer eigene mathematische Routinen, die nicht im Basic implementiert sind, in seine Programme einbinden. Dieses Kommando ruft ein Maschinenprogramm auf, dessen Startadresse vorher in den Speicherzellen 1 und 2 abgelegt wurde. Wer sich dieses Befehls nicht bedient, hat drei Zeropagespeicherzellen zur freien Verfügung, die er für eigene Maschinenprogramme nutzen kann. Die Syntax der USR-Funktion entspricht derjenigen normaler Basic-Funktionen wie zum Beispiel FRE, POS, SIN, COS, TAN etc.: A= USR (B) oder PRINT USR (B).

Der Vorteil gegenüber dem SYS-Befehl liegt in der Möglichkeit Parameter zu übergeben. In unserem Beispiel wird der Benutzerroutine die Variable B übergeben. Der errechnete Zahlenwert kann dann seinerseits wieder einer Variablen zugewiesen werden.

Wie kann man sich in einem Maschinenprogramm die eingegebene Zahl beschaffen? Nun, alle Basic-Variablen werden im Fließkommaformat abgespeichert, eine Ausnahme bilden nur die Integer-Variablen. Dabei werden die Zahlen vom Computer nach einem bestimmten Verfahren binär verschlüsselt, so daß eine Zahl mit acht Nachkommastellen und einem Exponenten in nur fünf Bytes Platz findet.

Zwischengespeichert wird das Ergebnis in besonderen Fließkomma-Akkumulatoren (Adresse 97-104). Mit Hilfe eines Unterprogramms aus dem Basic-Interpreter kann es von dort aus abgerufen werden, wobei es in eine Integerzahl (also ein Wert zwischen 0 und 65536) umgewandelt wird.

Aufgerufen wird die Routine mit JSR $D7F7. Der 2-Byte-Wert steht dann in den Zeropagestellen $14 und $15 (Hexadezimal) zur Verfügung. Da diese Adressen vom Interpreter oft benutzt werden, ist es ratsam die Zahlen in andere Register zu übertragen (in welche, darauf kommen wir später noch zu sprechen).

Das Gegenstück zu dem eben gezeigten Unterprogramm stellt die Routine »2-Byte in Fließkomma« dar. Das Low-Byte wird in das y-Register geladen, das höherwertige Byte muß in den Akku. Dann wird die Routine mit JSR $D391 gestartet, wobei die USR-Variable (in unserem Beispiel A) die Daten erhält.

Speicher ausgedeutet — die Basic-Zeiger

Adresse 43 — 56: Diese Adressen spielen, wie wir schon im ersten Teil gesehen haben, bei der Verwaltung von Basic-Programmen und -Variablen eine zentrale Rolle. Über sie erfolgt die Trennung zwischen Programm und den einzelnen Variablentypen. Auch für das Speichern und Laden von Programmen liefern sie die Basisdaten. Die einzelnen Funktionen können sie Tabelle 1 entnehmen.

Eine interessante Gebrauchsmöglichkeit ergibt sich durch die Memoryroutine. Dies ist ein kurzes Basic-Programm, welches die Aufgabe hat, den von Programmen, Variablen, Strings und Arrays belegten Speicherplatz festzustellen (Listing 1):
Wie man nach dem Starten des Programms sehen kann, erhält man die Werte, indem man die Zeiger — nachdem sie in Dezimalzahlen umgewandelt worden sind — voneinander subtrahiert. Den Speicherplatz, der durch Variablen belegt ist, erhält man beispielsweise durch Subtraktion des Hilfszeigers, »Beginn der Variablen« (45/46) von dem Zeigerpaar »Beginn der Arrays« (47/48).

Gerade bei stark limitiertem Speicherplatz (wie zum Beispiel in der Grundversion) kann dieses Hilfsprogramm Aufschluß über die momentane Speicherverteilung geben.

Eine weitere Verwendung ergibt sich durch geschicktes Manipulie-

ren der Speicherzellen, so daß es möglich wird, mehrere Programme gleichzeitig um Speicher unterzubringen.

Diese Aufgabe erfüllt das recht komfortable Programm »Basic-Switch« (Listing 2); zunächst aber die Grundlagen: Die besagten Zeiger stecken, wie bereits im ersten Teil unseres Kurses beschrieben, den Adreßbereich für Basic-Programme ab. Die ersten vier Bytes aus dem Zeropagekomplex (Adresse 43 bis 46) grenzen das Programm nach oben und unten ab. Die daran folgenden Variablen sind in ihrer Ausdehung ebenfalls durch ein Zeigerpaar (Adresse 55/56) limitiert. Der Bereich jenseits dieser Markierung ist für Basic tabu. Dort ist also Platz für Maschinenprogramme oder ähnliches, denn ein Überschreiben durch Basic-Variable ist nicht möglich.

Will man nun mehrere Basic-Programme im Speicher versammeln, so muß man sie nur durch verstellen der Zeiger eindeutig voneinander trennen.

Listing 1. »Memory Dump«

```
10000 REM *** MEMORY DUMP ***
10005 REM
10010 A=46:B=44:GOSUB10080:PRINT"PROGRAM
M  :";X
10020 A=48:B=46:GOSUB10080:PRINT"VARIABL
EN :";X
10030 A=50:B=48:GOSUB10080:PRINT"ARRAYS
   :";X
10040 A=56:B=52:GOSUB10080:PRINT"STRINGS
   :";X
10050 A=56:B=44:GOSUB10080:PRINT"SPEICHE
R  :";X
10060 PRINT"BYTES FREE:";FRE(0)
10070 END
10080 X=PEEK(A)*256+PEEK(A-1)-PEEK(B)*25
6-PEEK(B-1)
10090 RETURN
READY.
```

Wie in Bild 1 zu sehen ist, grenzen die Programmblöcke direkt aneinander. Unter Programmblock ist hier die Einheit von Programm und einem sich daran anschließenden Variablenbereich zu verstehen. Durch Umschalten der Zeiger wird dann immer der gewünschte Programmblock eingeblendet.

Nützlich kann diese Art der Speicheraufteilung sein, wenn man umfangreiche Programme erstellen möchte und es nötig wird, andauernd Hilfsprogramme (zum Beispiel Grafikeditor oder ähnliches) nachzuladen. Durch Basic-Switch kann hier viel Arbeit eingespart werden.

Diese Routine liegt wieder in zwei-

Listing 2. »Basic-Switch« (Basic-Lader)

```
10 REM **************************
20 REM ****                  ****
30 REM ****   BASICSWITCH    ****
40 REM ****                  ****
50 REM ****   BY C. SAUER    ****
60 REM ****                  ****
70 REM **************************
80 FORT=0TO286:READD:P=P+D:NEXT:IFP<>380
21THENPRINT"[illegible]FEHLER !!":END
90 POKE55,0:POKE56,PEEK(56)-2:CLR:A=PEEK
(56)
100 RESTORE:FORT=A*256TOA*256+286
110 READD
120 IFD=-1THEND=A
130 IFD=-2THEND=A+1
140 POKET,D
150 NEXT
160 PRINT"[illegible]FERTIG. START MIT":PRINT"[illegible]S
YS"A*256+259
170 PRINT"[illegible]AKTIVIERUNG MIT 'A'"
180 GETA$:IFA$=""THEN180
190 IFA$<>"A"THENEND
200 SYSA*256+259
210 DATA230,122,208,002,230
220 DATA123,032,121,000,201
230 DATA255,240,003,076,121
240 DATA000,032,115,000,201
250 DATA069,208,003,076,168
260 DATA-01,201,083,240,003
270 DATA076,008,207,032,155
280 DATA215,165,101,240,004
290 DATA228,251,144,003,076
300 DATA072,210,134,252,032
310 DATA133,-01,166,252,202
320 DATA138,010,010,010,168
330 DATA162,000,185,032,-02
340 DATA149,043,200,232,224
350 DATA004,208,245,162,000
360 DATA185,032,-02,149,055
370 DATA200,232,224,002,208
380 DATA245,165,252,133,250
390 DATA032,096,198,032,142
400 DATA198,032,115,000,240
410 DATA006,201,143,240,016
420 DATA208,245,160,001,177
430 DATA122,208,239,200,192
440 DATA002,208,247,076,116
450 DATA196,032,215,202,165
460 DATA122,164,123,032,030
470 DATA203,240,241,166,250
480 DATA202,138,010,010,010
490 DATA168,162,000,181,043
500 DATA153,032,-02,200,232
510 DATA224,004,208,245,162
520 DATA000,181,055,153,032
530 DATA-02,200,232,224,002
540 DATA208,245,096,032,133
550 DATA-01,032,115,000,032
560 DATA138,205,032,184,209
570 DATA165,253,133,043,165
580 DATA254,133,044,165,101
590 DATA024,101,253,133,253
600 DATA165,100,101,254,133
610 DATA254,165,251,133,252
620 DATA230,251,165,043,208
630 DATA002,198,044,198,043
640 DATA032,248,227,165,254
650 DATA205,132,002,144,016
660 DATA208,007,165,253,205
670 DATA131,002,144,007,169
680 DATA012,160,205,032,030
690 DATA203,165,253,133,055
700 DATA165,254,133,056,032
710 DATA068,198,165,252,133
720 DATA250,076,049,-01,162
730 DATA000,134,250,134,116
740 DATA232,134,251,134,252
750 DATA165,043,133,253,165
760 DATA044,133,254,169,076
770 DATA133,115,169,-01,133
780 DATA117,096
READY.
```

erlei Form ausgedruckt vor. Zum einen als Assemblerlisting für solche, die tiefer in das Programm einsteigen möchten (Listing 3). Und zum anderen als Basic-Lader. Das Programm ist grundsätzlich auf jeder Ausbaustufe lauffähig. Sinnvoll wird es jedoch erst bei größerem Speicher. Es ist in bewährter Weise über das Befehlswort PI in den Interpreter eingebunden.

Nachdem die Routine per SYS gestartet worden ist, (der Lader übernimmt auch diese Arbeit), muß der erste Programmbereich initialisiert werden, das heißt der Benutzer nimmt die größenmäßige Einteilung des Speichers vor. Man muß sich jedoch schon im voraus darüber im klaren sein, wie lang das Programm inklusive Variablen sein soll, denn eine nachträgliche Änderung ist nicht möglich.

Die Syntax für die Initialisierung lautet: π E »Länge«. Danach steht der Speicherbereich abzüglich zwei Bytes für Basic zur Verfügung. Der erste so eingerichtete Programmblock hat die Nummer 1, der zweite 2 und so fort. Nachdem ein zweiter Programmbereich eingerichtet wurde, kann mit π S »Nummer« zwischen den Blöcken umgeschaltet werden, wobei die Routine jedoch jedesmal die Variable löscht.

Um dem Benutzer die Orientierung zu erleichtern, druckt das Maschinenprogramm nach jedem Umschalten die ersten REM-Zeilen aus. Daher ist es ratsam, jedem Programm eine Kopfzeile mit den wichtigsten Informationen zu geben.

Nach dieser etwas umfangreichen Erläuterung kehren wir nun zur Beschreibung der Zeropage zurück.

Adresse 59,60: Diese Speicherzellen enthalten die laufende Zeilennummer eines Basic-Programms. Sollte es im Programmlauf unterbrochen werden, hat man hier die Möglichkeit, die Zeilennummer nachzulesen.

Tabelle 1. Die Adreßbelegung der Zeropage beim VC 20

Dezimal	Hexadezimal	Bemerkung
0 - 2	00 - 02	USR-Sprungvektor (Normal: JMP $D248)
3 - 4	03 - 04	Vektor für Unterprogramm 'Fließkomma nach Integer'
5 - 6	05 - 06	Vektor für Unterprogramm 'Integer nach Fließkomma'
7	07	Suchzeichen (sucht ':' oder Zeilenende)
8	08	Hochkommaflag
9	09	Spaltenspeicher beim TAB-Befehl
10	0A	LOAD/VERIFY Flag (0 = LOAD/1 = VERIFY)
11	0B	Eingabepufferzeiger (Anzahl der Elemente)
12	0C	Flag für Dim (enthält die laufende Variable)
13	0D	Variablentyp (0 = Numerisch/128 = String)
14	0E	Numerische Variable (0 = Fließkomma/128 Integer)
15	0F	Flag bei DATA und LIST
16	10	Flag für FN
17	11	Eingabeflag (0 = INPUT/64 = GET/152 = READ)
18	12	Vorzeichen bei ATN
19	13	Aktuelles Ein-/Ausgabegerät
20 - 21	14 - 15	Integerwert (zum Beispiel Zeilennummer oder Zwischenerg.)
22	16	Zeiger im Stringstapel
23 - 24	17 - 18	Zeiger auf den zuletzt verwendeten String
25 - 33	19 - 21	Stringstapel
34 - 37	22 - 25	Speicher für div. Hilfszeiger
38 - 42	26 - 2A	Speicherbereich bei einer Multiplikation
43 - 44	2B - 2C	Zeiger auf Beginn des Basic-Speicherbereichs
45 - 46	2D - 2E	Beginn der Variablen (= Ende des Programms)
47 - 48	2F - 30	Beginn der Arrays (= Ende der Variablen)
49 - 50	31 - 32	Zeiger auf Ende der Arrays
51 - 52	33 - 34	Zeiger auf Beginn der Strings
53 - 54	35 - 36	Hilfszeiger für Strings
55 - 56	37 - 38	Zeiger auf Basic-Ende
57 - 58	39 - 3A	Momentane Basic-Zeilennummer
59 - 60	3B - 3C	Vorherige Basic-Zeilennummer
61 - 62	3D - 3E	Zeiger auf lfd. Basic-Befehl (für CONT)
63 - 64	3F - 40	Momentane Zeilennummer für DATA
65 - 66	41 - 42	Momentane Adresse einer DATA-Zeile
67 - 68	43 - 44	Zeiger auf Element in DATA-Zeile und INPUT-Vektor
69 - 70	45 - 46	Momentaner Variablenname
71 - 72	47 - 48	Adresse der momentanen Variablen
73 - 74	49 - 4A	Zeiger für FOR/NEXT Variable
75 - 76	4B - 4C	Zwischenspeicher für Programmzeiger (bei SAVE)
77	4D	Speicher für Vergleichssymbole
78 - 79	4E - 4F	Zeiger für FN
80 - 83	50 - 53	Verschieden genutzter Speicherbereich (Hauptsächlich für Strings)
84 - 86	54 - 56	FN-Sprungvektor (ähnlich USR)
87 - 96	57 - 60	Feld für diverse arithmetische Zwecke (Arithmetik Akku #3 und #4)
97	61	Fließkommaakku #1 : Exponent
98 - 101	62 - 65	Fließkommaakku #1 : Matisse
102	66	Fließkommaakku #1 : Vorzeichen
103	67	Zeiger für Polynomauswertung
104	68	Überlauf von Akku #1
105	69	Fließkommaakku #2 : Exponent
106 - 109	6A - 6D	Fließkommaakku #2 : Mantisse

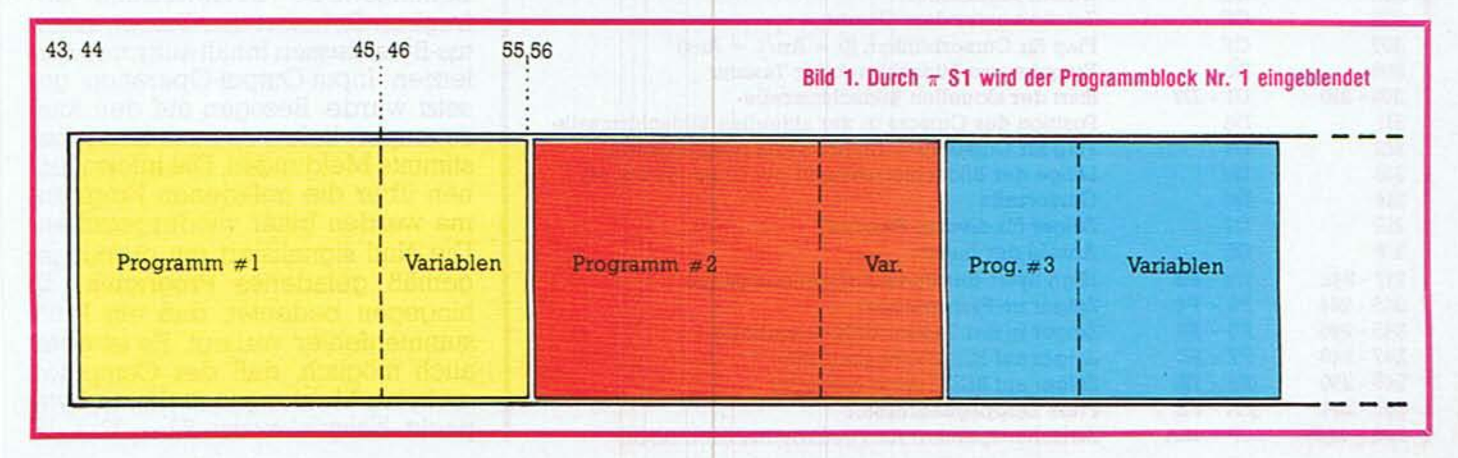


Bild 1. Durch π S1 wird der Programmblock Nr. 1 eingeblendet

110	6E	Fließkommaakku #2 : Vorzeichen
111	6F	Vergleichsbyte der Vorzeichen von Akku #1 und Akku #2
112	70	Rundung für Akku #1
113 - 114	71 - 72	Länge des Kassettenpuffers
115 - 138	73 - 8A	CHRGET-Routine
139 - 143	8B - 8F	RND-Wert als Fließkommazahl
144	90	Statusflag ST
145	9)	
150	96	Kassetten EOT (End of Tape) erhalten
151	97	Zwischenspeicher für Register
152	98	Anzahl der offenen Dateien
153	99	Eingabegerät (Normal = 0 : Tastatur)
154	9A	Ausgabegerät (Normal = 3 : Bildschirm)
155	9B	Paritätsbyte)
150	96	Kassetten EOT (End of Tape) erhalten
151	97	Zwischenspeicher für Register
152	98	Anzahl der offenen Dateien
153	99	Eingabegerät (Normal = 0 : Tastatur)
154	9A	Ausgabegerät (Normal = 3 : Bildschirm)
155	9B	Paritätsbyte vom Band (Prüfsumme)
156	9C	Flag für Byte erhalten (von Band)
157	9D	Flag für Direktmodus (= 128) oder Programm (= 0)
158	9E	Band: erster Teil - Prüfsumme
159	9F	Band: zweiter Durchlauf - Prüfsumme
160 - 162	A0 - A2	Interne Uhr (Stunde, Minute, Sekunde)
163	A3	Bitzähler für serielle Ausgabe
164	A4	Zähler für Band
165	A5	Startsynchronisation bei Kassetten
166	A6	Zeiger im Bandpuffer
167 - 171	A7 - AB	Flags für Schreiben/Lesen bei Band
172 - 173	AC- - AD	Zeiger auf Kassettenpuffer und für scrolling
174 - 175	Ae - AF	Zeiger auf Programmende bei LOAD
176 - 177	B0 - B1	Bandzeitkonstanten
178 - 179	B2 - B3	Startadresse des Bandpuffers
180	B4	Bitzähler für Band
181	B5	Band oder RS232: nächstes zu sendendes Bit
182	B6	Übertragungsspeicher für RS232
183	B7	Länge des Filenamens
184	B8	Logische Filenummer
185	B9	Sekundäradresse
186	Ba	Gerätenummer
187 - 188	BB - BC	Zeiger auf Filenamen
189	BD	Ein-/Ausgabespeicher (für seriell)
190	BE	Blockzähler für Band
191	BF	Puffer für serielle Ausgabe
192	C0	!Bandmotor Flag
193 - 194	C1 - C2	Startadresse für Ein-/Ausgabe
195 - 196	C3 - C4	Endadresse für Ein-/Ausgabe
197	C5	gedrückte Taste (momentaner Wert im Tastaturmatrix-code)
198	C6	Anzahl der gedrückten Tasten (im Tastaturpuffer)
199	C7	Bildschirm: negative Anzeige
200	C8	Zeiger auf Zeilenende bei Eingabe
201	C9	Cursorzeile
202	CA	Cursorspalte
203	CB	Welche Taste? (64 = keine)
204	CC	Cursorblinken
205	CD	Cursor Blinkzähler
206	CE	Zeichen unter dem Cursor
207	CF	Flag für Cursorblinken (0 = An/1 = Aus)
208	D0	Eingabe von Bildschirm/oder Tastatur
209 - 210	D1 - D2	Start der aktuellen Bildschirmzeile
211	D3	Position des Cursors in der aktuellen Bildschirmzeile
212	D4	Flag für Cursor (0 = Direkt, sonst programmiert)
213	D5	Länge der Bildschirmzeile (21 od. 42 od. 63 od. 84)
214	D6	Cursorzeile
215	D7	Zeiger für diverse Zwecke
216	D8	Anzahl der Inserts
217 - 242	D9 - F2	High Bytes der Bildschirmzeilenanfänge
243 - 244	F3 - F4	Zeiger im Farbspeicher
245 - 246	F5 - F6	Zeiger in der Tastaturdekodiertabelle
247 - 248	F7 - F8	Zeiger auf RS232 Eingabepuffer
249 - 250	F9 - FA	Zeiger auf RS232 Ausgabepuffer
251 - 254	FA - FE	Freie Zeropageadressen
255 (- 266)	FF - 10A	Zwischenspeicher für Fließkomma nach ASCII

Tabelle 1. Die Adreßbelegung der Zeropage beim VC 20 (Fortsetzung)

Adresse 63 — 66: Mit Hilfe dieser Zeiger läßt sich ein zeilennummerngesteuertes RESTORE realisieren. Die ersten zwei Bytes enthalten die momentane Zeilennummer für die DATA-Abfrage. Dies ist sehr praktisch, denn damit kann man jederzeit feststellen, in welcher DATA-Zeile man sich befindet. Das zweite Doppelbyte bildet die Adresse für die nächste DATA-Zeile. Man hat also zwischen Zeilennummer und der absoluten Adresse einer DATA-Zeile zu unterscheiden.

Das nun folgende kurze Maschinenprogramm (Listing 4 und 5) ermöglicht ein Zeilennummern-RESTORE. Man kann damit zwar nicht gezielt auf ein Element in einer DATA-Zeile zugreifen, aber zumindest auf eine bestimmte Zeile.

Was auch im Zusammenhang mit dem Maschinensprachkurs von Interesse sein kann, ist der Aufbau dieser kurzen Routine. Denn durch geschicktes Nutzen von Unterprogrammen aus dem ROM kann man nämlich viel Speicherplatz sparen. Aus diesem Grunde werde ich in einer späteren Folge Interpreter- und Betriebssystemunterroutinen näher beleuchten.

Das vorliegende Maschinenprogramm benötigt genau 43 Byte und liegt am Ende des verfügbaren Basic-Speichers. Die Start- und Endadresse wird vom Lader angegeben. Aufgerufen wird der Zeilenrestore mit: SYS »Startadresse«, »Zeilennumer«.

Die DATA-Zeiger werden dann auf die angegebene Zeile zurückgestellt, was sowohl im Direktmodus, als auch vom Programm aus erfolgen kann.

Zustandsbeschreibung: Das Statusflag

Adresse 144: Dieses Statusflag ist auch von Basic aus über die STATUS beziehungsweise ST-Anweisung abfragbar. Es liefert das Computerstatus-Byte, dessen Inhalt aufgrund der letzten Input-Output-Operation gesetzt wurde. Bezogen auf den Kassettenport liefert es nach Bild 2 bestimmte Meldungen. Die Informationen über die geladenen Programme werden binär wiedergegeben. Die Null signalisiert ein ordnungsgemäß geladenes Programm. 32 hingegen bedeutet, daß ein Prüfsummenfehler vorliegt. Es ist aber auch möglich, daß der Computer mehrere Meldungen in dieses Byte packt, beispielsweise 52 = 32 + 16

+ 4 : Hier wurde ein kurzer Block geladen, jedoch ist die Prüfsumme falsch und ein fataler Ladefehler liegt vor.

An dieser Stelle ist es angebracht, sich näher mit dem Aufzeichnungsverfahren zu beschäftigen (Bild 3).

So kommen Programme auf's Band

Jeder Abspeichervorgang beginnt mit dem Header. Dieser Kopf besteht aus dem Vorspann (das ist ein etwa acht Sekunden langer Pfeifton) und dem eigentlichen Programmkopf. Dieser enthält vier wichtige Informationen, nämlich über Programmtyp, Startadresse, Endadresse und Programmname. Diese Daten sind ebenfalls im Bandpuffer zu finden und können von dort abgerufen werden.

Das erste Byte (Adresse 828) gibt Auskunft über den Headertyp. Eine 1 zeigt an, daß es sich um ein Programm handelt, das verschoben geladen werden kann, also auch an eine andere Stelle, als die, von der aus es abgespeichert wurde. Das Gegenstück dazu ist die absolute Lademethode (Headertyp 3), die bereits in der ersten Folge meiner Abhandlung vorgestellt wurde. Gemeint ist LOAD"·",1,1. Die Sekundäradresse 1 signalisiert dem Computer, daß er das Proramm (unabhängig von den Zeigern 43, 44) wieder in den gleichen Adreßbereich laden soll. Um Verwechslungen vorzubeugen, ist es wichtig, Headertyp und Sekundäradresse zu unterscheiden. Für den Headertyp gibt es drei Möglichkeiten:

1: Laden mit Verschiebelader (die Anfangsadresse wird durch den Zeiger 43 und 44 bestimmt).
2: Ein File — also Daten aus Variablen — wurde abgespeichert. Die Unterscheidung ist wichtig, denn Daten können nicht mit LOAD geladen werden.
3: Ein Programm ist absolut zu laden.

Die nächsten vier Bytes geben die Anfangs- und Endadresse des geladenen Files an. Mit der Endadresse hat es eine besondere Bewandtnis. Sie wird nämlich nach korrektem Laden der Zeropage (Adresse 45, 46) übergeben. Bei Load Error geschieht dies nicht; PRINT FRE(0) zeigt dann die volle Bytezahl an, obwohl sich ein Programm im Speicher befindet. Man darf das Programm — das vielleicht nur einen geringfügigen Fehler hat —, dann nicht starten, weil die Variablen dieses überschreiben würden. Abhilfe schafft in diesem Fall

POKE 45,PEEK(831):POKE 46,PEEK (832):CLR.

Die Werte werden damit von »Hand« übertragen und ein normaler Programmablauf ist in den meisten Fällen wieder möglich.

Jetzt aber wieder zurück zu Bild 3: Nachdem der Vorspann und der Programmkopf vor einem Trennzeichen (dies ist ein ganz kurzer Pieps) auf Band geschrieben worden ist, wird der Header gleich noch einmal abgespeichert.

Dies ist eine Eigenheit des Commodore-Systems, das der Datensicherheit dient. Denn nachdem der Programmkopf 1 geladen hat, vergleicht er in einem zweiten Durchgang das bisher geladene (welches sich ja schon im Speicher befindet) mit Programmkopf 2. Eine Abweichung veranlaßt den Computer eine Fehlermeldung auszugeben, in ganz schweren Fällen wird der Ladevorgang gleich ganz unterbrochen. Diese Verfahrensweise gilt nicht nur für den Programmkopf, sondern auch für Programme und Daten — sie alle werden doppelt abgespeichert.

Nachdem also der Programmkopf erkannt worden ist, zeigt der Computer den Filenamen an, und das eigentliche Programm (oder die Daten) wird geladen. Ihnen ist wiederum ein kurzer Vorspann vorangestellt. Der Endblock besteht aus Endmarkierung und einem Nachspann. Er zeigt dem VC 20 das Ende des geladenen Programms an, womit der Ladevorgang beendet ist.

Nach diesem Exkurs zum Kassetenaufzeichnungsformat nun wieder zur Zeropage.

Die STOP-Taste mit Sonderfunktion

Adresse 145: Mit Hilfe dieser Speicherstelle kann man den Zustand der STOP- und der linken SHIFT-Taste abfragen:

Wert =
253: Linke SHIFT-Taste gedrückt
254: STOP-Taste gedrückt
255: Keine der beiden gedrückt

Wenn man das Low-Byte des STOP-Vektors (Adresse 808) auf 114

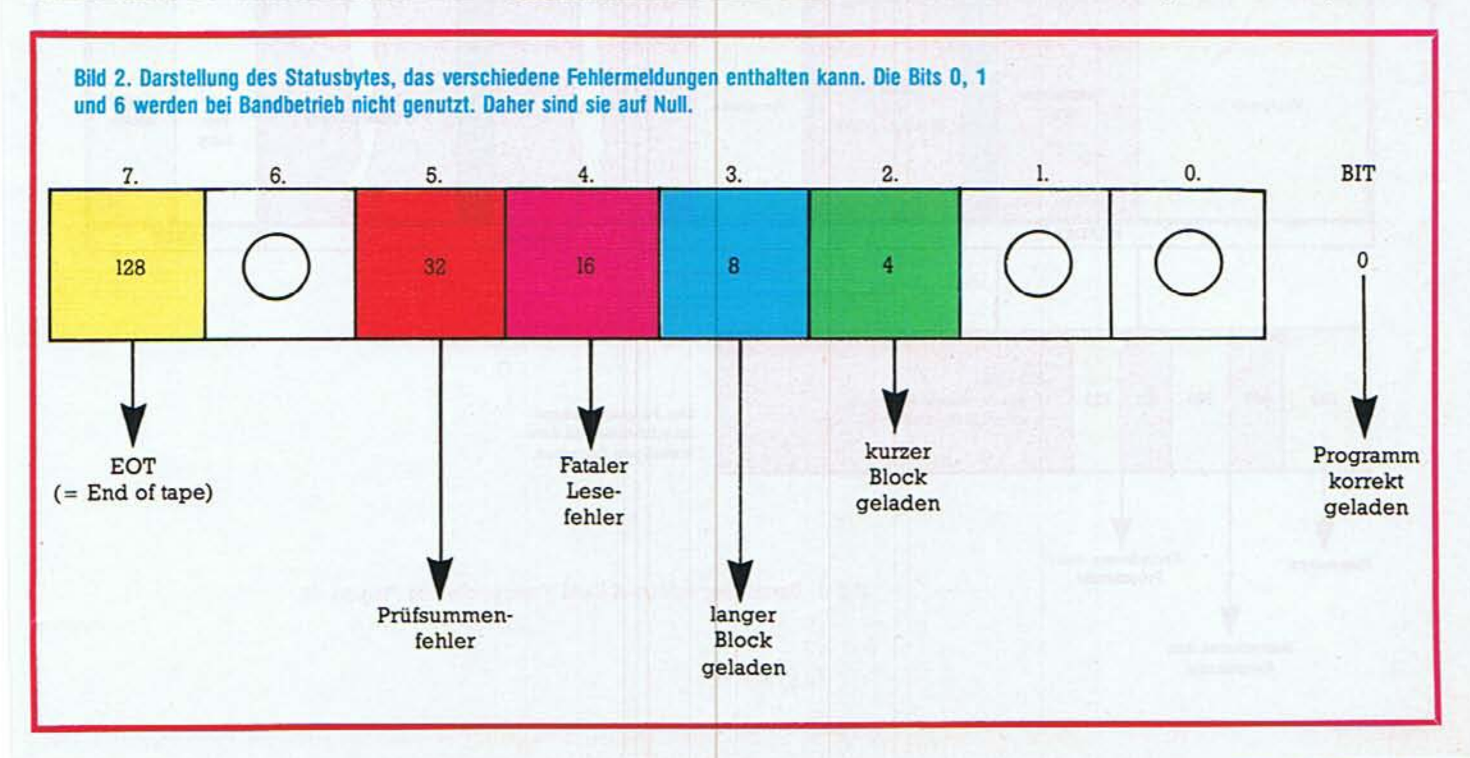


Bild 2. Darstellung des Statusbytes, das verschiedene Fehlermeldungen enthalten kann. Die Bits 0, 1 und 6 werden bei Bandbetrieb nicht genutzt. Daher sind sie auf Null.

— statt ursprünglich 112 — setzt, so ist man in der Lage, diese Taste von Basic aus abzufragen, ohne das laufende Programm anzuhalten. Ihr kann dann in der Routine eine besondere Funktion zugewiesen werden, beispielsweise Anhalten des Programmablaufs für eine bestimmte Zeit.

Programmiert oder direkt? Das ist hier die Frage

Adresse 157: Dieses Flag kann nur zwei Zustände annehmen: Entweder 0 oder 128. Ist der Inhalt der Speicherstelle Null, dann arbeitet der Computer gerade ein Programm ab; bei 128 befindet er sich im Direktmodus.

Diese Zeropageadresse kann dann von Bedeutung sein, wenn man eigene Basic-Kommandos definieren will (vergleiche Teil 1).

Zum Beenden einer solchen Routine gibt es zwei Möglichkeiten: Entweder kam der Programmbefehl aus dem Direktmodus. In diesem Fall springt man in die Interpreterschleife (JMP $C474) zurück, damit der Computer READY meldet. Kam das Kommando aber aus einem Basic-Programm, so muß man mit JMP $0079 in die CHRGOT-Routine zurückspringen, damit kein Bereitschaftszeichen ausgegeben wird. Mit Hilfe dieses Flags trifft man hier die Unterscheidung.

Nun wenden wir uns einem anderen Kapitel zu, der Cursorverwaltung. Über die Zeropage kann jederzeit die Cursorposition festgestellt werden. Die Speicherstelle 211 zeigt die Spalte, 214 die Zeile, in der sich der Blinker derzeit befindet. Korrekte Werte liefern die zwei Adressen aber nur innerhalb eines Programms.

Interessanter ist die umgekehrte Verfahrensweise, nämlich das Setzen des Cursors an eine beliebige Bildschirmposition. Dazu müssen allerdings vier Adressen geändert werden. Glücklicherweise nimmt uns eine Unterroutine aus dem Betriebssystem diese Arbeit ab.

Gewußt wo — der Cursor

In Maschinensprache wird das X-Register mit dem Zeilenwert, das Y-Register mit dem Spaltenwert geladen. Danach ruft man mit JSR $E50C das Unterprogramm auf.

Von Basic aus werden die zwei CPU-Register über die Adressen 781 und 782 geladen (warum das so ist, sehen wir in der nächsten Folge): POKE 781,»Zeile«: POKE 782,»Spalte«: SYS 58636.

Diese Verfahrensweise ist, wie man sieht, wesentlich komfortabler als das Hantieren mit den Steuerzeichen.

Eine andere Zeropageadresse kann uns zu einer weiteren Erleichterung verhelfen. Jeder kennt das Problem, den Cursor innerhalb von Anführungszeichen (»Gänsefüßchen«) zu bewegen. Es werden nur Steuerzeichen ausgedruckt, eine Bewegung des Zeigers findet nicht statt.

Dieses Manko läßt sich durch eine kurze Maschinenroutine beheben:

```
LDA #$00      ;
lösche Hochkommaflag
STA $D4       ;
speichere es ab
JMP $FEAD     ;
springe in die NMI-Routine
```

Ausgelöst wird diese Routine durch Drücken der RESTORE-Taste. Dazu muß vorher der NMI-Vektor (eine nähere Beschreibung nächstes Mal) auf den Beginn unserer Routine gestellt werden. Da die sehr kurz ist, legen wir sie im Kassettenpuffer ab. Dort ist sie solange sicher verwahrt, bis man etwas laden oder abspeichern will. Hier zunächst der Einzeiler, der die Routine in den Bandpuffer generiert:

```
10 FOR T=828 TO 8 34: READD: POKE T,D:
NEXT: DATA 169, 0, 133, 212, 76, 173, 254: POKE 792, 60: POKE 793, 3
```

Arbeitet man im Hochkommamodus, so kann man durch Drücken der RESTORE-Taste diesen Betriebszustand abschalten, der Cursor kann dann wieder ganz normal bewegt werden.

Abschließend möchte ich noch sagen, welche Zeropageadressen

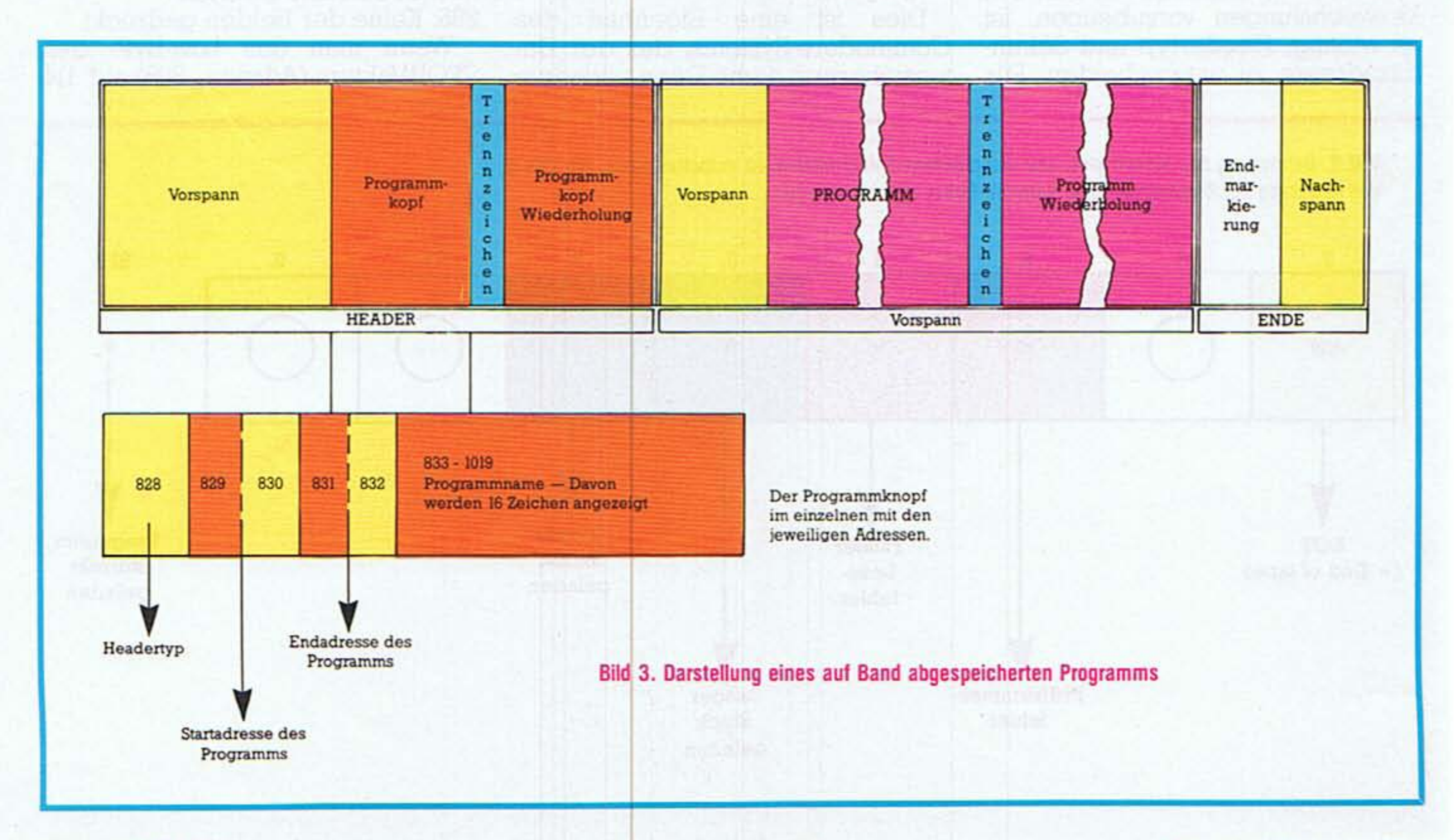


Bild 3. Darstellung eines auf Band abgespeicherten Programms

ohne Komplikationen von eigenen Maschinenroutinen genutzt werden können, was im hohen Maße vom Verwendungszweck abhängt.

Nicht verändert werden dürfen solche Speicherstellen, die vom Betriebssystem gebraucht werden, also die Adressen 192 bis 244 und 160 bis 162. Hat das Maschinenprogramm keine Verbindung zu Basic, so können alle Speicherstellen zwischen 0 und 138 überschrieben werden, weil sich hier nur Basic-Parameter befinden. Eng wird der Platz für die sogenannten Utilities — also für Basic-Ergänzungen oder Hilfsprogramme. Will man Daten nur zwischenspeichern, kann man sie in den Bereich zwischen Adresse 147 bis 159 packen. Dort sind sie bis zum nächsten Lade- oder Abspeichervorgang sicher.

Dauerhaft können Daten nur am Anfang und am Ende der Zeropage abgelegt werden. Dazu gehören Adresse 0 bis 2 und 245 bis 254.

Genau diese werden aber auch von käuflichen Basic-Erweiterungen gebraucht, wodurch es zu Komplikationen kommen kann. Also Vorsicht, erst durch vorheriges Probieren testen, ob man sie benutzen darf!

Damit möchte ich für heute schließen. In der nächsten Folge werden schwerpunktmäßig der Tastaturpuffer, die Basic-Vektoren und die Kernalvektoren behandelt.

(Christoph Sauer/ev)

Listing 3 finden Sie auf Seite 164

Listing 4. »Zeilen-RESTORE« (Basic-Lader)

```
10 REM ************************************
*********
20 REM ****
    ****
30 REM **** ZEILENNUMMERNGESTEUERTES RES
TORE ****
40 REM ****
    ****
50 REM ****       1984 BY CHRISTOPH SAUER
    ****
60 REM ****  HUBERTUSSTR.14 / 8000 MUEN.
 19   ****
70 REM ****
    ****
80 REM ************************************
*********
90 FORT=0TO47:READD:S=S+D:NEXT:IFS<>5219
THENPRINT"FEHLER":END
100 POKE55,0:POKE56,PEEK(56)-1:CLR:A=PEE
K(56)
110 RESTORE:FORT=A*256TOA*256+47
120 READD
130 POKET,D
140 NEXT
150 PRINT"[illegible]FERTIG. START MIT":PRINT"[illegible]S
YS"A*256
160 DATA032,121,000,201,044
170 DATA240,003,076,008,207
180 DATA032,115,000,032,138
190 DATA205,032,247,215,165
200 DATA020,133,063,165,021
210 DATA133,064,032,019,198
220 DATA032,248,200,165,095
230 DATA164,096,056,233,001
240 DATA032,036,200,076,116
250 DATA196,116,196
READY.
```

Listing 5. »Zeilen-RESTORE« (Assembler-Darstellung)

```
1D00 JSR $0079  ; CHRGOT, LFD. BASICZEICHEN
1D03 CMP #$2C   ; KOMMA ?
1D05 BEQ $1D0A  ; JA, DANN VERZWEIGEN
1D07 JMP $CF08  ; SONST 'SYNTAX ERROR'
1D0A JSR $0073  ; CHRGET, NAECHSTES ZEICHEN
1D0D JSR $CD8A  ; AUSDRUCK HOLEN
1D10 JSR $D7F7  ; IN 2-BYTE FORMAT WANDELN
1D13 LDA $14    ; UEBERTRAGUNG DER ZEILENNR. (LB)
1D15 STA $3F
1D17 LDA $15    ; HB UEBERTRAGEN
1D19 STA $40
1D1B JSR $C613  ; ADRESSE DER DATAZEILE BERECHNEN
1D1E JSR $C8F8  ; NAECHSTE DATAZEILE SUCHEN
1D21 LDA $5F
1D23 LDY $60
1D25 SEC
1D26 SBC #$01
1D28 JSR $C824  ; RESTORE DURCHFUEHREN
1D2B JMP $C474  ; RUECKSPRUNG INS BASIC
```

Listing 3. »Basic-Switch« (Assembler-Darstellung)

```
BASICSWITCH

****************** CHRGET ERGAENZUNG
1C00 INC $7A     ; CHRGET FORTFUEHREN
1C02 BNE $1C06
1C04 INC $7B
1C06 JSR $0079
1C09 CMP #$FF    ; TOKEN = ← ?
1C0B BEQ $1C10   ; JA, DANN VERZWEIGEN
1C0D JMP $0079   ; SONST ZURUECK INS BASIC
1C10 JSR $0073   ; CHRGET, NAECHSTES ZEICHEN
1C13 CMP #$45    ; 'E' FUER ERGAENZUNG
1C15 BNE $1C1A   ; NEIN/ DANN WEITER
1C17 JMP $1CA8   ; SONST IN ERGAENZUNGSROUTINE SPRINGEN
1C1A CMP #$53    ; 'S' FUER SWITCH ?
1C1C BEQ $1C21   ; JA, DANN VERZWEIEN
1C1E JMP $CF08   ; SONST 'SYNTAX ERROR'
****************** ←S FUER SWITCH
1C21 JSR $D79B   ; BYTEWERT (0-255)HOLEN
1C24 LDA $65
1C26 BEQ $1C2C   ; WENN WERT=0, DANN FEHLER
1C28 CPX $FB
1C2A BCC $1C2F   ; ? WENN WERT<ANZAHL DER PROGRAMME, DANN WEITER
1C2C JMP $D248   ; SONST FEHLERMELDUNG
1C2F STX $FC     ; PROGRAMMNUMMER ABSPEICHERN
1C31 JSR $1C85   ; ADRESSEN 43-56 IN ZWISCHENSPEICHER UBERNEHMEN
1C34 LDX $FC     ; EINGABEWERT * 8
1C36 DEX
1C37 TXA
1C38 ASL
1C39 ASL
1C3A ASL
1C3B TAY
1C3C LDX #$00    ; ZWISCHENSPEICHER IN REGISTER 43-46 SCHREIBEN
1C3E LDA $1D20,Y
1C41 STA $2B,X
1C43 INY
1C44 INX
1C45 CPX #$04
1C47 BNE $1C3E
1C49 LDX #$00    ; ZWISCHENSPEICHER IN REGISTER 55-56 SCHREIBEN
1C4B LDA $1D20,Y
1C4E STA $37,X
1C50 INY
1C51 INX
1C52 CPX #$02
1C54 BNE $1C4B
1C56 LDA $FC     ; LFD. PROGRAMMNR.
1C58 STA $FA     ; GLEICH ALTER PROGRAMMNR.
1C5A JSR $C660   ; BASIC BEFEHL CLR
***************** REM- ZEILA AUSDRUCKEN
1C5D JSR $C68E   ; CHRGET ZEIGER (7A,7B) AUF BASICSTART
1C60 JSR $0073   ; CHRGET, NEUES KOMANDO HOLEN
1C63 BEQ $1C6B   ; BEIM ZEILENENDE VERZWEIGEN
1C65 CMP #$8F    ; 'REM' CODE ?
1C67 BEQ $1C79   ; JA, DANN AUSDRUCKEN
1C69 BNE $1C60   ; SONST ZURUECK
1C6B LDY #$01
1C6D LDA ($7A),Y; ZEILENENDE AUCH PROGRAMMENDE ?
1C6F BNE $1C60   ; NEIN, DANN ZURUECK
1C71 INY
1C72 CPY #$02
1C74 BNE $1C6D
1C76 JMP $C474   ; PROGRAMMENDE, DANN IN BASIC DIREKTMODUS
1C79 JSR $CAD7   ; LEERZEILE AUSGEBEN
1C7C LDA $7A     ; AKKU   AUF L-BYTE DES STRINGS
1C7E LDY $7B     ; Y REG. AUF H-BYTE DES STRINGS
1C80 JSR $CB1E   ; STRING (=REM ZEILE) AUSDRUCKEN
1C83 BEQ $1C76   ; ZURUECK INS BASIC
****************** REGISTER ABSPEICHERN
1C85 LDX $FA     ; ALTE PROGRAMMNR. * 8
1C87 DEX
1C88 TXA
1C89 ASL
1C8A ASL
1C8B ASL
1C8C TAY
1C8D LDX #$00    ; ZEIGER 43-46 IN ZWISCHENSPEICHER UEBERNEHMEN
1C8F LDA $2B,X
1C91 STA $1D20,Y
1C94 INY
1C95 INX
1C96 CPX #$04
1C98 BNE $1C8F
1C9A LDX #$00    ; ZEIGER 55-56 IN ZWISCHENSPEICHER
1C9C LDA $37,X
1C9E STA $1D20,Y
1CA1 INY
1CA2 INX
1CA3 CPX #$02
1CA5 BNE $1C9C
1CA7 RTS         ; ZURUECK
****************** ←E FUER ERGAENZUNG
1CA8 JSR $1C85   ; ZEIGER (43-56) IN ZWISCHENSPEICHER
1CAB JSR $0073   ; CHRGET, NEUES ZEICHEN HOLEN
1CAE JSR $CD8A   ; NUMERISCHEN AUSDRUCK HOLEN
1CB1 JSR $D1B8   ; FLIESKOMMA IN 2-BYTE FORMAT WANDELN
1CB4 LDA $FD     ; HOECHSTE ADRESSE DES 1. PROGRAMMS =
1CB6 STA $2B     ; NIEDRIGSTE ADRESSE DES 2. PROGRAMMS
1CB8 LDA $FE
1CBA STA $2C
1CBC LDA $65     ; EINGEGEBENER WERT
1CBE CLC
1CC1 STA $FD     ; UND ABSPEICHERN (L-BYTE)
1CC3 LDA $64
1CC5 ADC $FE     ; ADDITION DER H-BYTES
1CC7 STA $FE     ; UND ABSPEICHERN
1CC9 LDA $FB     ; ANZAHL DER VERWALTETEN PROGRAMME HOLEN
1CCB STA $FC     ; UND DER LFD. PROGRAMMNR. GLEICHSETZEN
1CCD INC $FB     ; ? ANZAHL DER PROGRAMME UM 1 ERHOEHEN
1CCF LDA $2B     ; BASICANFANG UM 1 ZURUECK
1CD1 BNE $1CD5
1CD3 DEC $2C
1CD5 DEC $2B
1CD7 JSR $E3F8   ; 0 AN DEN BASICANFANG UND BA. UM EINS ERHOEHEN
1CDA LDA $FE     ; UEBERPRUEFUNG, OB DIE NEUE BASICENDADRESSE
1CDC CMP $0284   ; UEBERHAUPT NOCH IM SPEICHERBEREICH LIEGT
1CDF BCC $1CF1
1CE1 BNE $1CEA
1CE3 LDA $FD
1CE5 CMP $0283
1CE8 BCC $1CF1   ; LIEGT ADRESSE IM SPEICHERBEREICH, DANN WEITER
1CEA LDA #$0C    ; SONST FEHLERMELDUNG (L-BYTE DER FM LADEN)
1CEC LDY #$CD    ; H-BYTE LADEN
1CEE JMP $CB1E   ; STRING 'REDO FROM START' AUSDRUCKEN
1CF1 LDA $FD     ; BERECHNETES BASICENDE
1CF3 STA $37     ; DEM ECHTEN GLEICHSETZEN
1CF5 LDA $FE
1CF7 STA $38
1CF9 JSR $C644   ; BASIC ROUTIENE NEW
1CFC LDA $FC     ; LFD. PROGRAMMNR. =
1CFE STA $FA     ; ALTE PROGRAMMNR
1D00 JMP $1C31   ; EINSPRUNG IN DIE SWITCH ROUTINE
****************** INITIALISIERUNG
1D03 LDX #$00
1D05 STX $FA     ; ALTE PROGRAMMNR. INITIALISIEREN
1D07 STX $74     ; CHRGET AENDERN
1D09 INX
1D0A STX $FB     ; ANZAHL DER PROGRAMME UND
1D0C STX $FC     ; LFD. PROGRAMMNR. AUF1
1D0E LDA $2B
1D10 STA $FD     ; HIMEM ZEIGER INITIALISIEREN
1D12 LDA $2C
1D14 STA $FE
1D16 LDA #$4C    ; CHRGET AENDERN
1D18 STA $73
1D1A LDA #$1C
1D1C STA $75
1D1E RTS
```

Listing 3. »Basic-Switch« (Assembler-Darstellung, Schluß)

Mit 4 Baud



Man stelle sich vor: Der Nachbar morst empfängt die Zeichen. Auf dem Bildschirm Nicht möglich? In Sindelfingen schon. Lichtzeichen. Datenfernübertragung

Bild links: Von Balkon zu Balkon. Die Gegenstelle (Pfeil) ist immerhin 200 m entfernt. Daneben in Großaufnahme das Empfangsgerät

Die Nachbarn haben sich inzwischen daran gewöhnt. Immer wenn es dunkel wird, blinken auf zwei Balkons Morsezeichen auf. Dann wissen die Anwohner, daß die beiden 19jährigen Freunde Mark Schneider und Dietmar Schwuchow wieder mit ihren Computern spielen.

Die Computer-Karriere der beiden begann vor zwei Jahren: Wie viele andere auch studierten sie Handbücher, tippten Listings ab, schrieben schließlich eigene Spiele und »Nutzprogramme«. Im Frühjahr brach bei beiden der Spieltrieb durch. »Wir waren es leid, immer getrennt mit unseren Computern zu arbeiten. So kamen wir auf die Idee, sie als Kommunikationsmittel einzusetzen. Ein Kabel oder ein Akustikkoppler war uns jedoch zu langweilig«, erklärten Dietmar und Mark, die beide gerade ihr Abitur erfolgreich hinter sich gebracht haben.

Die Freunde wohnen so nah beieinander, daß sie sich vom Balkon aus zuwinken können. Daher erschien es ihnen am einfachsten, Lichtimpulse für die Datenübertragung zu nutzen. Schnell einigten sich die Elektronik-Fans über diesen Weg. »Klar wußten wir von ersten Moment an, daß wir damit keine professionelle Datenfernübertragung machen können. Was uns reizt, ist vielmehr das Basteln und Spielen. Unter diesem Vorzeichen sehen wir auch unser selbstgebautes Lichttelefon«. Für Mark und Dietmar zählt das Außergewöhnliche ihrer Sende- und Empfangseinrichtung mehr als ein tatsächlicher Nutzen.

Vier Monate brauchten sie, um die Idee in die Tat umzusetzen. Jetzt spielen sie zusammen »Schiffchen versenken« — jeder an seinem Computer. Der übernimmt mittlerweile »artig« die Ver- und Entschlüsselung der Daten sowie das »Morsen« bei der Übertragung. Vier Bit huschen in jeder Sekunde von Balkon zu Balkon und übermitteln die Züge an den jeweils anderen Computer.

Zunächst machten sich die Computer-Freaks über die Hardware her. Die beiden Computer, ein C 64 und ein Sinclair ZX Spectrum, mußten durch entsprechende Elektronik mit je einer Lampe als Sender und einem Fotowiderstand als Empfänger verbunden werden.

Ins Innenleben des Computers wurde nicht eingegriffen

Da Mark und Dietmar nicht unbedingt in das Innenleben der Computer eingreifen wollten, nutzten sie die vorhandenen Anschlüsse. Der Tonausgang dient zum Ansteuern der Lampe: beim C 64 ist dies der Audio-Ausgang und beim ZX Spectrum der Recorderanschluß. Den Signalempfang übernimmt der jeweilige Joystick-Port.

Es war von vornherein vorgesehen, nur bei Dunkelheit zu übertragen. Trotzdem — die erste Schaltung erwies sich bei den Testläufen als zu unempfindlich. Mit einer verbesserten Elektronik war dieses Problem schnell gelöst.

Nachdem sich die beiden Freunde auf einen gemeinsamen Code geeinigt hatten, kam der große Moment: der erste richtige Übertragungsversuch. Jedem Bustaben wurde ein Fünf-Bit-Code zugeordnet. Es handelte sich um eine serielle Übermittlung, wobei die Lampe bei gesetztem Bit kurz aufblinkte. Das Senden begann mit einem Synchronisationssignal, um damit den empfangenden Computer auf den Übertragungstakt einzustellen. Da Basic für diese Zwecke nicht präzise genug arbeitet, kamen Mark und Dietmar auf die Idee, den internen Zeitzähler zu nutzen. Es funktionierte, die Daten wurden im Sekundenrhythmus übertragen — jede Sekunde ein Bit.

Ganz zufrieden waren die Elektronik-Tüftler noch nicht. Bei mehr als 20 Zeichen gab es noch Ungenauigkeiten in der Übertragung. Außerdem war ein Baud (Bit pro Sekunde) doch ein bißchen zu langsam. Fazit: Die Software wurde umgeschrieben, so daß die Geschwindigkeit jetzt vier Baud beträgt. Auch bei größeren Datenmengen fällt nur selten ein gesendetes Bit über den Balkon — die Information kommt zuverlässig beim anderen an.

über den Balkon

…mit einer Taschenlampe und Ihr Computer zeigt er Ihnen an, was Ihr Nachbar will. Zwei Computer senden und empfangen von Balkon zu Balkon.



Die beiden Elektronikbastler

Derzeit arbeiten Mark und Dietmar an der nächsten Verbesserung ihrer Datenfernübertragung. Die Auswertung von zwischengespeicherten Daten ist momentan erst möglich, nachdem alle Bits im Computer angekommen sind. Geplant ist ein Maschinenprogramm, das die empfangenen Zeichen sofort decodiert und ausdruckt. Die Lust am Basteln hält weiter an. Das nächste Projekt — ein Plotter aus Fischer-Technik-Bauteilen — steht kurz vor der Vollendung. Aber auch den »Ernst des Lebens« will sich — zumindest Mark — mit Computern versüßen. Bereits am Gymnasium hat er Lehrer »gespielt«. Jetzt versucht er, sein Computer-Wissen in der Volkshochschule »loszuwerden«.

(M. Schneider/ D. Schwuchow/kg)

Ablauf der Datenübertragung

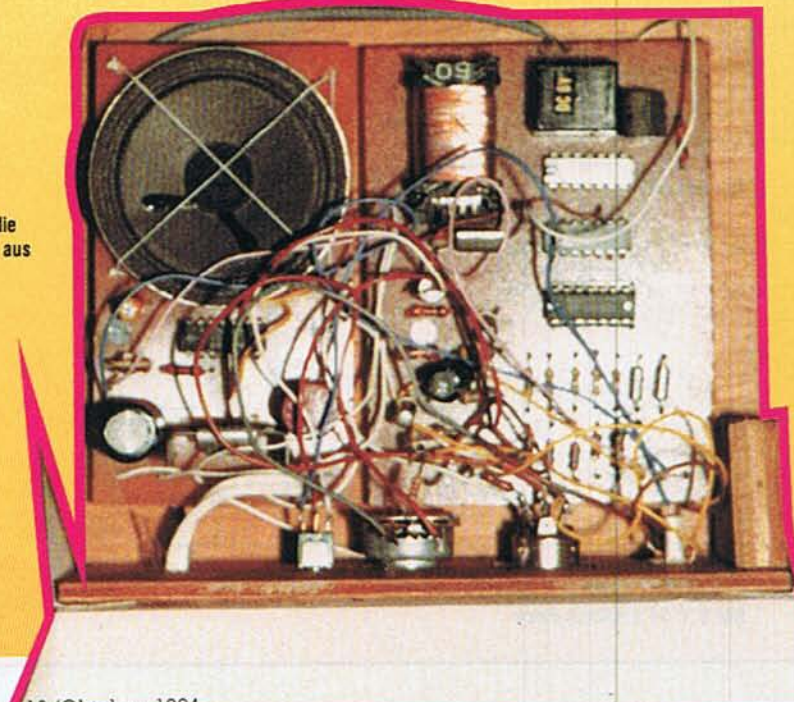
Die Datenübermittlung beginnt wie es sich gehört: Der Sender »klingelt« zunächst beim Empfänger. Dazu setzt er ein Blinksignal von ½ Hz, das bedeutet 2 Sekunden Licht an, 2 Sekunden aus und so weiter. Der Empfänger muß seine Bereitschaft signalisieren. Kennzeichen: einmal 4 Sekunden lang aufblinken. Wird dieses sogenannte Bereitschaftssignal erkannt, schaltet der vorher blinkende Computer auf Texteingabe um und beginnt zu senden. Der »angeklingelte« Computer stellt nach dem Abschicken des Bereitschaftssignals sofort auf Empfang um. Damit ist die Richtung der Informationsübermittlung festgelegt. Sie wechselt jetzt regelmäßig ohne vorheriges »Klingeln«.

Jeder Sendevorgang beginnt mit einem Synchronisationssignal — einem Aufblinken von 3 Sekunden Dauer. Eine Sekunde später folgen die Kontrollbytes: Zweimal fünf Bit im Viertelsekundenrhythmus. Damit wird dem empfangenden Computer mitgeteilt, wieviele Zeichen die Nachricht beinhaltet und wie oft demnach die Empfangsschleife durchlaufen werden muß, bevor er decodieren darf. Das Synchronisationssignal dauert drei Sekunden, dann folgt eine Sekunde lang Sendepause. Erst jetzt wird die eigentliche Nachricht geschickt, ebenfalls im Viertelsekundentakt.

Die Übertragungsrichtung wechselt am Ende der Mitteilung automatisch und der vorherige Empfänger startet seinen Sendevorgang mit dem Synchronisationssignal.

Wie kann der Empfänger dem Sender mitteilen, daß er nur »Schrott« mitbekommt, wenn die Übertragung immer nur auf eine Richtung festgelegt ist? Dafür gibt es eine Maschinen-Code-Routine.

Falls der Empfänger während der Übertragung einen Fehler feststellt, so drückt er einfach eine Error-Taste, beim C 64 ist dies die obere linke Pfeiltaste. Damit schaltet er seinen Computer auf »Texteingabe«, was bedeutet, er kann jetzt senden und dem anderen mitteilen, daß etwas falsch angekommen ist. Das Synchronisationssignal dauert dann — zur Unterscheidung von normaler, fehlerfreier Übermittlung — allerdings 10 Sekunden. Obwohl der andere Computer in dem Moment noch eifrig weitersendet



So sieht die Hardware aus

Mit 4 Baud über den

(er hat ja keine Ahnung, daß etwas nicht richtig rüberkommt) erkennt er ein Dauersignal von mehr als 5 Sekunden. Nach der Warnung schaltet er auf Empfang, um die Fehlermeldung des anderen aufzunehmen.

Hardwarebeschreibung

Computer und ein Teil der Elektronik sind im Zimmer untergebracht — der Rest des sogenannten Lichttelefons steckt in einem Sende- und Empfangskästchen auf dem Balkon. Bild 3 zeigt, was drinnen und was draußen steht.

Mark und Dietmar müssen bei der Datenübertragung eine Strecke von etwa 200 m überwinden. Das erfordert ein relativ intensives Lichtsignal, damit die Information überhaupt beim anderen ankommt. Eine 12V/10W-Lampe erbringt die notwendige Leistung.

Als Spannungsquelle für die Lampe dient ein ausrangierter (Rennbahn-)Trafo. Die 12V des Trafos sind außerdem mit einem Spannungsregler des Typs 7805 verbunden, der für eine konstante 5V-Spannung der zusätzlichen Hardware sorgt (Bild 1).

Die Sende- und Empfangskästchen sind mit Schirmklemmen am Balkongeländer befestigt. Nimmt man den Regenschutz — aus Pappe und aufgeklebter Aluminiumfolie — ab, so wird ein quaderförmige Holzgehäuse von 7 x 7 x 12 cm frei. In die Vorderwand wurde ein Loch für die Linse gesägt, die wiederum fest in einem Papprohr sitzt. Das Sendeelement (Lampe) und Empfangsteil, der lichtempfindliche Widerstand (LDR), müssen im Brennpunkt der Linse sein. Da diese fest eingebaut ist, wird eine genaue Einstellung nur durch Verschieben der Platine erreicht, auf die die Bauteile gelötet sind (Bild 6).

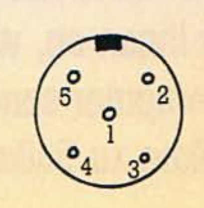


Bild 2. Pinbelegung des DIN-Steckers

1: GND (Ground-Masse)
2: L+ (12 V)
3: Signal von LDR
4: +5 V
5: Testsignal für LED



Bild 3. Die Verbindung zwischen drinnen und draußen

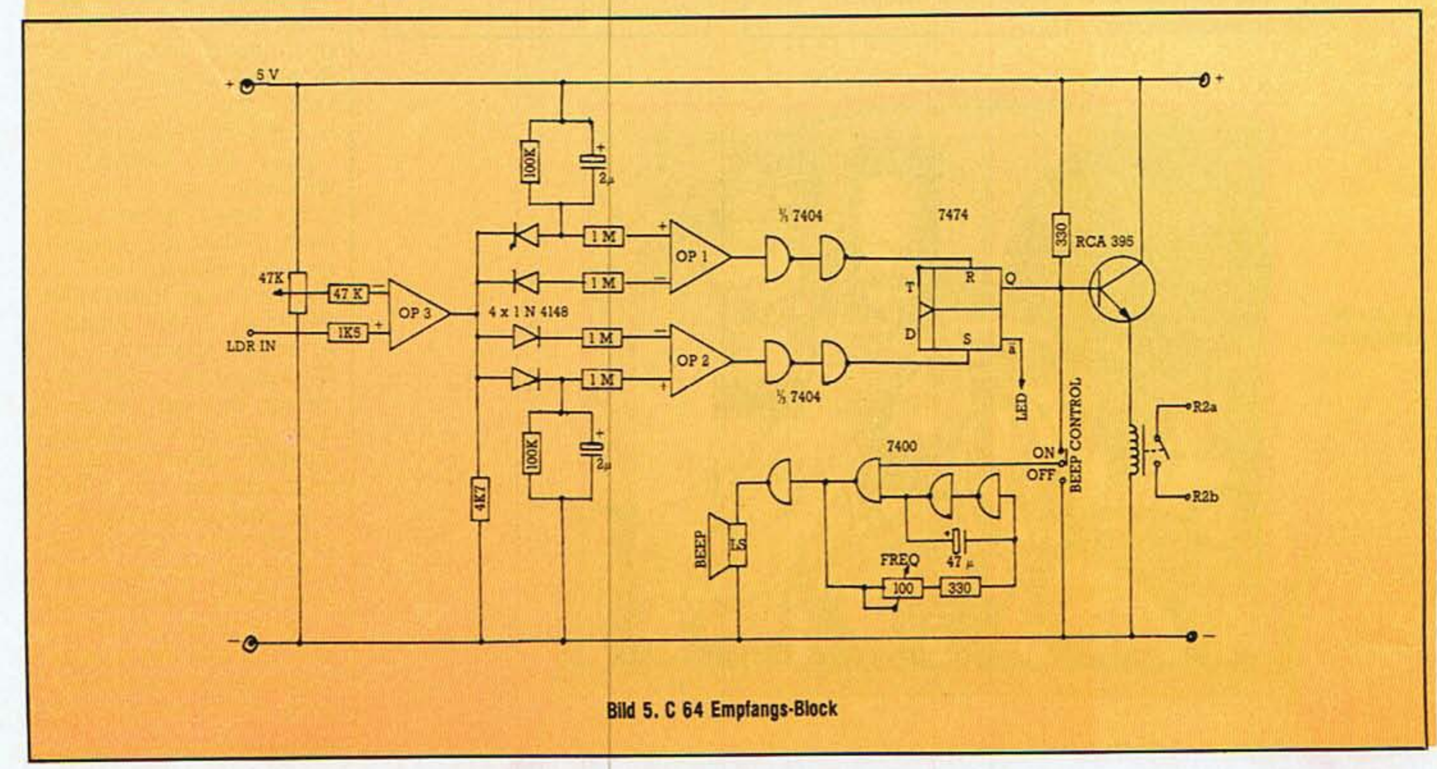


Bild 5. C 64 Empfangs-Block

Balkon

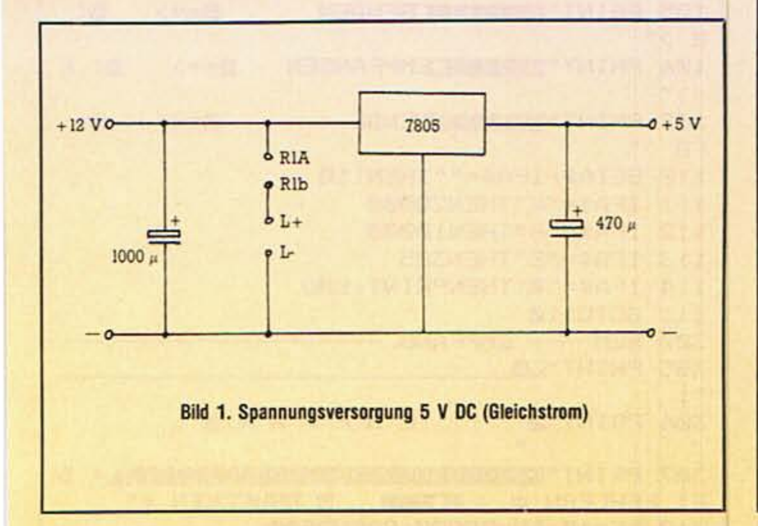


Bild 1. Spannungsversorgung 5 V DC (Gleichstrom)

Die Kästchen auf dem Balkon sind über ein vierpoliges abgeschirmtes Kabel mit der Elektronik im Zimmer verbunden. Das Ende des Kabels bildet ein DIN-Stecker, mit der in Bild 2 dargestellten Stiftbelegung.

Das Senden funktioniert nach einem einfachen Prinzip: Beim Morsen wird für logisch »1« am Audioausgang des Computers ein Tonsignal abgegeben, das durch die Schaltung (Bild 4) verstärkt und digitalisiert wird. Als Folge davon schließt das Relais und die Lampe im Sendekästchen leuchtet auf.

Beim Empfangen (Bild 5) vergleicht der Operationsverstärker (OP3) die

Zeichencodierung

Code	Tafel 1	Tafel 2
11111	Space	0
00001	A	1
00010	B	2
00011	C	3
00100	D	4
00101	E	5
00110	F	6
00111	G	7
01000	H	8
01001	I	9
01010	J	(
01011	K	)
01100	L	.
01101	M	&
01110	N	+
01111	O	—
10000	P	*
10001	Q	/
10010	R	.
10011	S	,
10100	T	:
10101	U	;
10110	V	>
10111	W	<
11000	X	=
11001	Y	?
11010	Z	!
11011	Tafel 2	—
11100	—	—
11101	—	—
11110	—	—

Die Zeichen aus Tafel 2 werden als Zwei-Byte-Code verschlüsselt. Das Byte »11011« dient als »Schaltbyte« zu dieser Tafel. Die Zeichen »Komma« und »Doppelpunkt« werden aus eingabetechnischen Gründen als »■« beziehungsweise als »[« eingegeben.

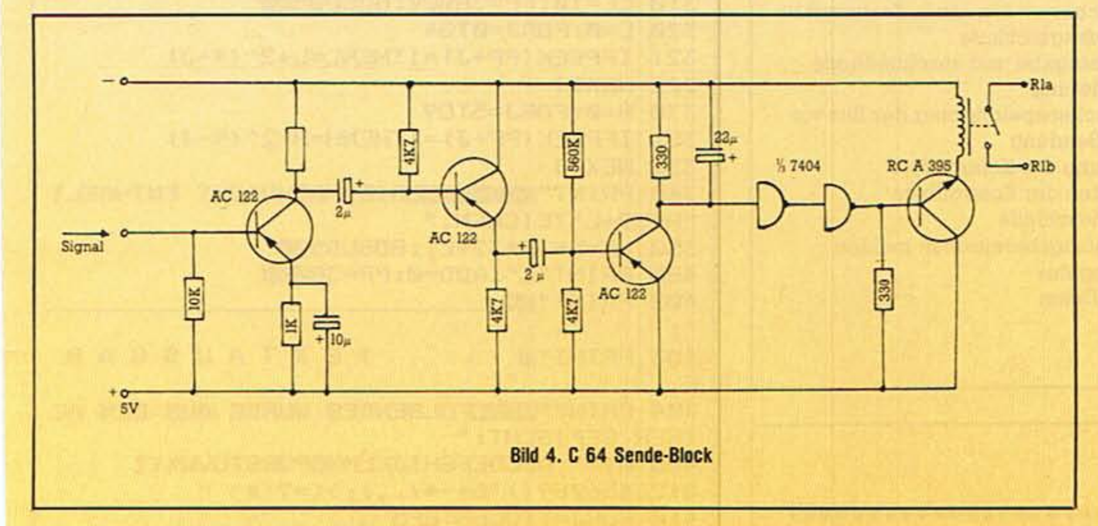
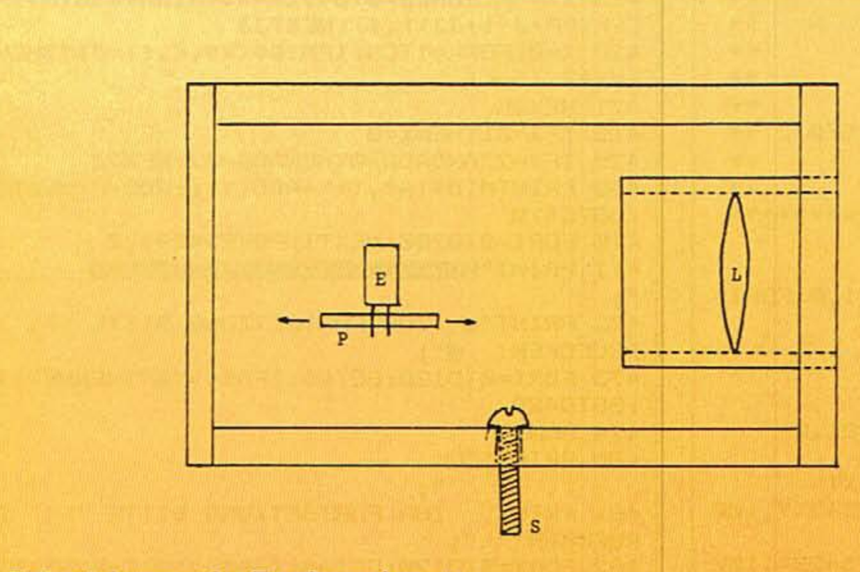


Bild 4. C 64 Sende-Block

Signale von beiden Spannungsteilern, die durch den lichtempfindlichen Widerstand (LDR) mit dem 100 KOhm-Widerstand und dem Regler 47 KOhm gebildet werden. Dieser Verstärker ist auch für die hohe Empfindlichkeit verantwortlich. Die nachfolgende Schaltung ist für langsame Helligkeitsschwankungen unempfindlich. Lediglich bei Lichteinfall setzt sie ein Flipflop und nimmt es bei Dunkelheit zurück. Dieses Flipflop steuert das Relais für den Joystick-Eingang an. Darüber hinaus bietet es eine doppelte Kontrolle beim Empfang: sowohl auf dem Balkon beim Einstellen der Kästchen über die LED als auch im Zimmer durch den BEEP.



E: Sendelampe oder Empfangselement
P: Verschiebbare Platine
L: Linse ∅ 35 mm, Brennweite 7 cm
S: Befestigungsschraube

Bild 6. Aufbau des Sende- und Empfangskästchens

Erläuterungen zum Lichttelefon-Listing:

Zeilen	6-31	Allgemeine Vorbereitungen Einlesen der MC-Daten, Setzen der Tonregister
	40-42	Speicherstelle 828 wird auf ihren Inhalt abgefragt. Daraufhin wird entsprechend zu »Empfang« oder »Senden« verzweigt. Diese Konstruktion entstand aus dem Wunsch, alle gesetzten Variablen durch »RUN« zu löschen und nicht einzeln.
	100-115	Hauptmenü
	300-340	Zunächst werden hier die beiden Kontrollbyte, die dem Empfänger die Länge der Nachricht angeben, empfangen. Dies ist nötig, da das eigentliche Empfangsprogramm die empfangenen Bits nur zwischenspeichert und somit kein Endsignal erkennen kann.
	400-483	Decodierung der beim Empfang ab Speicherstelle 38000 zwischengespeicherten Bits sowie Textausgabe
	500-650	Empfangsschleife
	10000-10140	Texteingabe und anschließende Codierung Zwischenspeicherung der Bits vor der Sendung
	10200-10300	Bildung der Kontrollbyte
	12000-12330	Senden der Kontrollbyte
	12400-12650	Sendeschleife
	14000-14034	Empfangsbereitschaft melden
	20000-20120	»Klingeln«
	21000-21010	MC-Daten

```
0 REM ***********************************
      **                              **
1 REM **    L I C H T T E L E F O N    **
      **                              **
2 REM **    VON M.SCHNEIDER UND        **
      **         D.SCHUCHOW            **
3 REM **    FRIEDRICH-EBERT STR.5/072 **
      **                              **
4 REM **    7032 SINDELFINGEN          **
      ***********************************
5 :
6 IX=828
7 PRINT"♥":POKE53280,0:POKE53281,0:FORI=
49152TO49172:READA:POKEI,A:NEXT
8 GOSUB14000
10 DIMX$(255)
11 IX=828
13 PRINT"♥":POKE53280,0:POKE53281,0
20 POKE56,145:POKE650,128
28 FORI=54272TO54296:POKEI,0:NEXT
30 POKE54272,1:POKE54273,0:POKE54277,100
:POKE54278,100:POKE54276,17
31 POKE54279,1:POKE54280,0:POKE54284,100
:POKE54285,100:POKE54283,33
40 IFPEEK(IX)=3THENPOKEIX,0:GOTO300
42 IFPEEK(IX)=7THEN10000
100 PRINT"♥X_______________________________
";
101 PRINT"■          H A U P T M E N U E
  "
104 PRINT"■■■■■■■■'KLINGELN'     ■==>     ■<
 K >"
105 PRINT"■■■■■■■■SENDEN         ■==>     ■<
S >"
106 PRINT"■■■■■■■■EMPFANGEN      ■==>     ■< E
 >"
107 PRINT"■■■■■■■■ENDE           ■==>     ■<
F8 >"
110 GETA$:IFA$=""THEN110
111 IFA$="K"THEN20000
112 IFA$="S"THEN10000
113 IFA$="E"THEN305
114 IFA$="■"THENPRINT:END
115 GOTO110
300 REM---- EMPFANG -------------------
305 PRINT"♥X_______________________________
";
306 PRINT"■        E M P F A N G
  "
307 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■* B
EI FEHLERN ■ _ ■■■■■    ■ DRUECKEN *"
310 LE=10:PP=38000:GOSUB500
320 L=0:FORJ=0TO4
321 IFPEEK(PP+J)=1THENL=L+2^(4-J)
322 NEXTJ
330 H=0:FORJ=5TO9
331 IFPEEK(PP+J)=1THENH=H+2^(9-J)
332 NEXTJ
340 PRINT"■■■■■■■■DIE NACHRICHT ENTHAELT
"H*32+L"ZEICHEN."
350 LE=5*(H*32+L):GOSUB500
400 PRINT"♥":ADD=0:PP=38000
402 PRINT"■X_______________________________
";
403 PRINT"■          T E X T A U S G A B
E      "
404 PRINT"■■■FOLGENDES WURDE AUS DEM AE
THER GEFISCHT:"
408 A$=" ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789()'&+-*/.,:;><=?!#^ "
410 FORJ=1TOLESTEP5
415 X$="":FORJJ=0TO4:X$=X$+RIGHT$(STR$(P
EEK(PP+J-1+JJ)),1):NEXTJJ
420 X=0:FORK=1TO5:IFMID$(X$,K,1)="1"THEN
X=X+2^(5-K)
421 NEXTK
422 IFX=31THENX=0
425 IFX=27ANDADD=0THENADD=32:NEXTJ
430 PRINTMID$(A$,1+X+ADD,1);:ADD=0:NEXTJ
:GOTO470
470 FORI=0TO700:NEXTI:POKE54296,0
471 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■X_______
";
472 PRINT"■ ZUR FORTSETZUNG BITTE 'F1'
DRUECKEN!  ■";
473 FORI=0TO120:GETA$:IFA$<>"■"THENNEXTI
:GOTO480
474 RUN10
480 PRINT"□□
        ";
481 PRINT"  ZUR FORTSETZUNG BITTE 'F1' D
RUECKEN!  ";
482 FORI=0TO120:GETA$:IFA$<>"■"THENNEXTI
:GOTO471
483 RUN10
500 REM---- EMPFANGSSCHLEIFE -----------
510 IF(PEEK(56320)AND16)=0THEN518
512 IFPEEK(197)=57THENPOKEIX,7:RUN10
```

```
514 GOTO510
518 PRINT"                              A
 E U F T"
520 IF(PEEK(56320)AND16)=0THEN520
530 TI$="000000"
540 IFTI<54THEN540
550 Z$=""
560 FORL=1TOLE
562 IFPEEK(197)=57THENPOKEIX,7:RUN10
565 TI$="000000"
570 IF(PEEK(56320)AND16)=0THENA=TI:GOTO6
00
580 IFTI<14THEN570
590 POKEPP-1+L,0:NEXTL:RETURN
600 IF(PEEK(56320)AND16)=0THEN600
610 B=TI
620 X=(12-B+A)/2
630 POKEPP-1+L,1:TI$="000000"
640 IFTI<X-6THENGOTO640
650 NEXTL:RETURN
10000 REM---- SENDEN -------------------
10001 PRINT"  ________________________________
";
10002 PRINT"                S E N D E N
     "
10005 PRINT"    IHR TEXT?"
10006 PRINT" BITTE NACH JEDER ZEILE    R
ETURN                       DRUECKEN"
10010 A$="":FORI=1TO5:INPUTA$:T$=T$+A$:A
$="":NEXTI
10011 IFLEN(T$)=0THENPOKEIX,0:RUN10
10020 PP=38000:POKE54296,15
10030 B$="1111100001000100001100100000101
001100011101000010010101001011011100"
10032 B$=B$+"0110101110011111000010001l0
0101001110100101011011010111111000"
10034 B$=B$+"1100111010110111110011101111
1101111100000"
10036 C$="0123456789()'&+-*/. [;><=?!#^
"
10050 ADD=0:FORI=1TOLEN(T$)
10060 M$=MID$(T$,I,1)
10070 IFM$=" "THENX$=MID$(B$,1,5):GOTO10
100
10080 IFASC(M$)>64ANDASC(M$)<91THENX$=MI
D$(B$,(ASC(M$)-64)*5+1,5):GOTO10100
10085 FORII=1TOLEN(C$)
10086 IFMID$(C$,II,1)=M$THENX$="11011"+M
ID$(B$,(II-1)*5+1,5):ADD=ADD+1:GOTO10100
10087 NEXTII:X$="00000"
10100 FORJ=1TOLEN(X$)
10110 POKEPP,(ASC(MID$(X$,J,1))-48)
10120 PP=PP+1
10130 NEXTJ:X$(I)=X$
10140 NEXTI
10200 REM---- LAENGE FESTSTELLEN -------
10205 L=LEN(T$)+ADD
10206 PRINT"  DIE NACHRICHT ENTHAELT"L"Z
EICHEN."
10210 H=INT(L/32):L=L-H*32
10300 L$=MID$(B$,(L*5)+1-160*(L=0),5)+MI
D$(B$,(H*5)+1-160*(H=0),5)
12000 REM---- KONTROLLBYTES SENDEN -----
12100 PRINT"
* SENDUNG LAEUFT * "
12105 POKE54273,100:TI$="000000"
12106 IFTI$<>"000003"THEN12106
12107 IFPEEK(IX)=7THENIFTI$<>"000010"THE
N12107
12108 POKE54273,0:IFPEEK(IX)=7THENPOKEIX
,0
12110 TI$="000000"
12120 IFTI<60THEN12120
12200 FORI=1TOLEN(L$)
12210 TI$="000000"
12215 SYS49152:IFPEEK(54280)<>0THENPOKEI
X,3:FORI=0TO2000:NEXT:RUN10
12220 IFMID$(L$,I,1)="0"THEN12300
12230 IFTI<3THEN12230
12235 POKE54273,100
12240 IFTI<11THEN12240
12245 POKE54273,0
12300 IFTI<15THEN12300
12320 NEXTI
12330 FORI=0TO3000:NEXTI
12400 REM---- HAUPTSENDESCHLEIFE -------
12405 POKE54273,100:TI$="000000"
12406 IFTI$<>"000003"THEN12406
12407 POKE54273,0
12420 TI$="000000"
12430 IFTI<60THEN12430
12510 FORZR=38000TO38000+5*LEN(T$)+5*ADD
-1
12520 TI$="000000"
12525 SYS49152:IFPEEK(54280)<>0THENPOKEI
X,3:FORI=0TO2000:NEXT:RUN10
12530 IFPEEK(ZR)=0THEN12600
12540 IFTI<3THEN12540
12545 POKE54273,100
12550 IFTI<11THEN12550
12555 POKE54273,0
12600 IFTI<15THEN12600
12620 NEXTZR
12650 POKEIX,3:RUN10
14000 REM---- BEREITSCHAFT MELDEN ------
14001 PRINT"           EMPFANGSBEREITSCH
AFT MELDEN (J/N)?"
14010 GETA$:IFA$="J"THEN14020
14011 IFA$<>"N"THEN14010
14012 RETURN
14020 PRINT"           * MELDUNG WIRD GE
SENDET *"
14030 POKE54296,15:POKE54273,100:TI$="00
0000"
14032 IFTI$<>"000004"THEN14032
14034 POKE54273,0:POKEIX,3:RETURN
20000 REM---- KLINGELN -----------------
20001 PRINT"  ________________________________
";
20002 PRINT"              ' K L I N G E L N
'
              "
20010 POKE54296,15
20020 TI$="000000"
20030 POKE54273,100
20040 IFTI$<>"000002"THEN20040
20050 POKE54273,0:TI$="000000"
20052 IFTI$<>"000001"THEN20052
20054 TI$="000000"
20060 IF(PEEK(56320)AND16)=0THEN20100
20062 IFTI$<>"000001"THEN20060
20070 GOTO20020
20100 PRINT"     MARK IST JETZT EMPFANGS
BEREIT!"
20110 TI$="000000"
20112 IFTI$<>"000004"THEN20112
20120 GOTO10000:REM SENDEN
21000 REM---- DATEN FUER MC-ROUTINE ----
21010 DATA173,0,220,201,111,208,13,232,1
38,201,20,208,7,162,0,169,152,141,8,212,
96
READY.
```

Listing »Lichttelefon« (Schluß)



Die Anzahl der eingegangenen Unterprogramme und deren unterschiedliche Thematik machte eine Bewertung natürlich nicht einfach. Wenn Ihr Programm diesmal nicht vorgestellt wird, so bedeutet das keine Abqualifikation. Damit Sie als Leser aber auch in den Genuß umfangreicherer Unterprogramme kommen, werden wir gelegentlich zusätzlich zu unserem »Unterprogramm des Monats« andere interessante Einsendungen veröffentlichen.

Der Autor dieses Programms ist ein waschechter Berliner. Jacques Effenberg ist 35 Jahre alt und von Beruf Maschinenschlosser. Im März 1983 kaufte er sich einen C 64. Nachdem er sich ein knappes Jahr mit Basic versuchte, stieg er auf Assemblerprogrammierung um, weil er mit Maschinensprache viele Probleme besser lösen konnte. Zum Programm selbst:

Das Programm Maske hat die Aufgabe, die Programmierung von Bildschirmmasken zu vereinfachen. Der erste Befehl erstellt horizontale und vertikale Linien. Die an den Schnittstellen benötigten Zeichen werden vom Programm ermittelt und eingesetzt. Der zweite Befehl löscht vom Benutzer definierte Felder in der Bildschirmmaske. Der dritte Befehl dient dazu, Text an bestimmte Stellen des Bildschirms zu drucken. Das Maschinenprogramm liegt im Speicherbereich von 49152 bis 50539. Eigene Fehlermeldungen und das Listen der Basic-Zeile in der der Fehler vorkommt, erleichtern dem Benutzer die Arbeit.

Parameterübergabe, Fehlermeldungen

1) Linien zeichnen

Der Befehl für Linien zeichnen wird in Basic wie folgt programmiert:

SYS 49152, Richtung, Zeile, Spalte, Länge, Farbe

1a) Richtung

Für eine horizontale Linie geben Sie eine 0 und für eine vertikale Linie eine 1 ein. Bei einer Zahl größer als 1 meldet sich der Computer mit der Fehlermeldung:

Richtung Wert zu groß (0 bis 1) in:

und listet die Zeile Ihres Basic-Programms.

1b) Zeile

Die Zeilenangabe liegt im Bereich von 1 bis 25. Bei Werten von kleiner als 1 oder größer als 25 lautet die Fehlermeldung:

Zeilenwert kleiner 1 oder größer 25 in:

und die Basic-Zeile wird gelistet.

1c) Spalte

Die Werte für die Spaltenangabe liegen im Bereich von 1 bis 40 und Werte kleiner als 1 oder größer als 40 ergeben die Fehlermeldung:

Spaltenwert kleiner 1 oder größer 40 in:

sowie eine Anzeige der Zeile.

1d) Die Länge kann im Bereich von 1 bis 40 für horziontale Linien und für vertikale Linien im Bereich von 1 bis 25 liegen. Werte, die nicht in diesem Zahlenbereich liegen, führen zu der Fehlermeldung:

Länge Linie zu klein oder zu groß in:

Wollen Sie zum Beispiel eine horizontale Linie zeichnen, mit dem Spaltenwert 20 und geben als Länge einen Wert von 22 an, so führt das zur Meldung:

Spalte + Länge Linie größer 40 in:

da der Bereich von 40 möglichen Spalten überschritten wird.

Eine vertikale Linie zum Beispiel ab Zeile 10 und mit der Länge 17 würde die Fehlermeldung:

Zeile + Länge Linie größer 25 in:

ergeben, da das Ergebnis von 10+17 den möglichen Bereich von 25 Zeilen überschreitet. In allen drei Beispielen erfolgt ein LISTen der Basic-Zeile.

1e) Farbe

Für die Farbwerte würden Zahlen im Bereich von 0 bis 15 in Frage kommen. Werte größer als 15 führen zu keiner Fehlermeldung.

2) Felder löschen

Um bestimmte Bereiche auf dem Bildschirm zu löschen, müssen Sie den Befehl:

SYS 50083, ab Zeile, bis Zeile, Spalte, Länge

eingeben.

2a) ab Zeile, bis Zeile

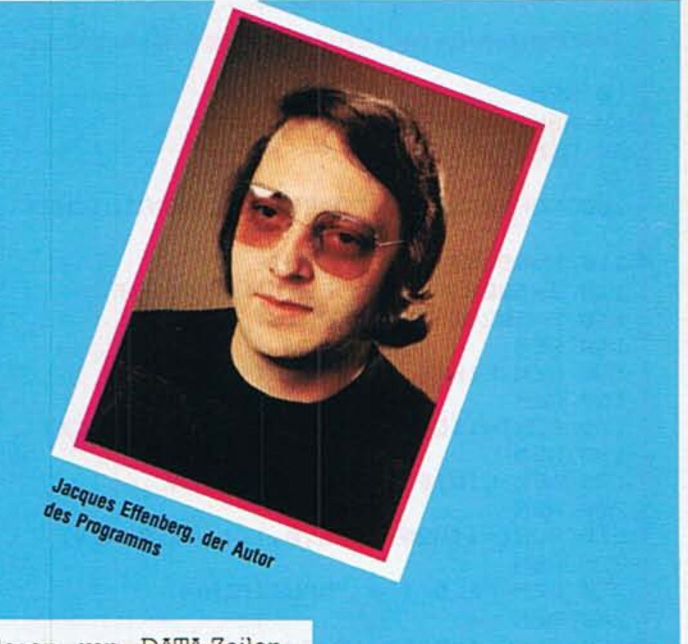
Wollen Sie zum Beispiel die ersten fünf Zeilen auf Ihrem Bildschirm komplett löschen, müßte der Befehl wie folgt aussehen:

SYS 50083,1,5,1,40

also ab Zeile 1 bis Zeile 5, ab Spalte 1 und 40 Zeichen lang. Ist nur eine Zeile zu löschen, dann ist der Wert ab Zeile gleich dem Wert bis Zeile.

Geben Sie zum Beispiel für ab Zeile 10 und bis Zeile 2





Jacques Effenberg, der Autor des Programms

ein, so erfolgt die Fehlermeldung:

Zeile 1 ist größer als Zeile 2 in:
sowie ein Listen der Basic-Zeile.

2b) Zahlenbereich

Für die Bereichsangabe gelten die Werte aus Absatz 1, sowie bei Überschreitungen auch die Fehlermeldungen.

3) Print

Für die gezielte Ausgabe eines Textes auf dem Bildschirm lautet der Befehl:

SYS 50242, Zeile, Spalte, Farbe, "Text"

3a) Zeile, Spalte, Farbe

Alle unter Absatz 1 beschriebenen Zahlenbereiche für die Angabe bei Zeile, Spalte, Farbe gelten auch hier. Ebenso treffen auch die dort erwähnten Fehlermeldungen bei einer Bereichsüberschreitung zu.

3b) "Text"

Hier kann all das stehen, was Sie auf dem Bildschirm ausgeben wollen.

Ein Beispiel:
SYS 50242,1,20,2,"Probe"
es wird in diesem Fall in der ersten Zeile ab Spalte 20 das Wort »Probe« in roter Farbe auf dem Bildschirm ausgegeben. Sie könnten auch vorher im Programm festlegen, daß D$ = "Probe" ist, schreiben dann SYS 50242,1,20,2, D$ und erhalten das gleiche Ergebnis.

Hatten Sie vor diesem Befehl eine grüne Cursorfarbe, dann wird, nach dem Ausgeben des Wortes »Probe« in roter Farbe, Ihre Cursorfarbe wieder grün sein.

4) Erklärungen zum DATA-Programm

4a) Programmablauf

In Zeile 1000 werden nach der Dimensionierung der Variablen P die 116 Prüfsummen für das Maschinenprogramm in P(I) eingelesen. Jede Prüfsumme ist das Ergebnis einer Addition von zwölf DATA-Werten. Die Variable SP addiert die Prüfsummen. Zeile 1130 überprüft die Gesamtsumme und gibt bei einer Differenz die Meldung »Fehler Prüfsumme« aus. Das Programm wird beendet.

Das Maschinenprogramm wird ab Zeile 1140 eingelesen und gespeichert. Die Variablen S und SI addieren und ZI zählt die DATA-Werte.

Ist ZI = 12 wird in Zeile 1160 die Variable S mit der jeweiligen Prüfsumme P (PI) verglichen. Ist S ungleich P (PI), verzweigt das Programm zur Zeile 1210.

In Zeile 1190 wird SI mit der Gesamtsumme aller DATAs verglichen. Bei einer Differenz ergibt sich ein Fehler in der letzten DATA-Reihe. Es kommt zur Ausgabe der Meldung »Fehler letzte Reihe DATA«.

Sind alle Werte richtig eingegeben, wird in Zeile 1200 nach der Meldung »OK« das Programm beendet. Beim Einlesen von DATA-Zeilen notiert sich der Interpreter die abzuarbeitende Zeilennummer in den Speicherstellen 63 und 64 der Zeropage. War in Zeile 1160 die Summe von 12 DATA-Werten und die Prüfsumme P (PI) ungleich, so wird in Zeile 1210 die Zeilennummer in der der Fehler auftrat in Z$ festgehalten.

Dann wird der Bildschirm gelöscht und in Zeile 1240 die Meldung »Fehler in Zeile« Z$ ausgegeben. In den Zeilen 1250 bis 1260 wird der Befehl »LIST« und die Zeilennummer in den Tastaturpuffer gePOKEt. Dazu kommt in Zeile 1270 noch der Befehl »RETURN« und in der Speicherstelle 198 wird dem Interpreter mitgeteilt, wieviel Stellen im Tastaturpuffer belegt sind.

Nach dem Befehl »END« in Zeile 1280 und der Ausgabe von »READY« befindet sich der Computer in einer Warteschleife, in der auch die Speicherstelle 198 abgefragt wird. Da der Wert in 198 größer als 0 ist, arbeitet der Interpreter jetzt den Tastaturpuffer in folgender Reihenfolge (zum Beispiel bei Fehler in Zeile 1290) ab:

Speicher 631 Wert 76 = L
Speicher 632 Wert 105 = i
(L + Shift i = Kurzzeichen für LIST)
Speicher 633 Wert 49 = 1
Speicher 634 Wert 50 = 2
Speicher 635 Wert 57 = 9
Speicher 636 Wert 48 = 0
Speicher 637 Wert 13 = Return

also LIST 1290 und die Zeile mit dem Fehler wird gelistet.

4b) Programmverbindung

Haben Sie alle DATA-Werte richtig eingegeben und der Computer hat sich nach einiger Zeit mit »OK« wieder gemeldet, so können Sie jetzt die Zeilen 1000 bis 1280 löschen. Dann geben Sie die folgende Zeile ein:

1280 FOR I = 49152 TO 50539:READ X : POKE I, X:NEXT:NEW

und speichern das Programm ab.

Wenn Sie das Programm in Zukunft benutzen, dann wird nach dem Einlesen des Maschinenprogramms in den Speicherbereich 49152 bis 50539 das DATA-Programm gelöscht, und Sie haben den ganzen Basic-Speicher für Ihr eigenes Programm zur Verfügung.

Sind Sie Besitzer eines Diskettenlaufwerkes, dann können Sie in der Zeile 1280 statt »NEW« den Befehl LOAD "...",8 schreiben, und das DATA-Programm lädt nach dem Einlesen der DATA-Zeilen Ihr eigenes Programm gleich nach.

Zwischen "..." muß der Name Ihres Programms stehen. Wenn Sie diese Variante wählen, dann muß aber die erste Zeile Ihres eigenen Programms wie folgt aussehen:

10 POKE 45, PEEK (174) : POKE 46, PEEK (175) : CLR

Durch diese Lösung wird das DATA-Programm auch gelöscht und Ihr Programm wird gleich nachgeladen und startet sich selbst.

(Jacques Effenberg/gk)

Dieses Demo-Programm zeigt anschaulich die Möglichkeiten des Unterprogramms

```
10 REM-----------------------------------
20 REM MASKE DEMO-PROGRAMM
30 REM-----------------------------------
40 :
100 POKE53280,0:POKE53281,0:PRINTCHR$(14
7)
110 L=49152   :REM LINIE
120 S=50083   :REM FELD LOESCHEN
130 P=50242   :REM PRINT
140 REM
150 REM LINIE 40 * VERTIKAL
160 REM
170 FORI=1TO40:SYSL,1,1,I,25,13:NEXT
180 REM
190 REM LINIE 25 * HORIZONTAL
200 REM
210 FORI=1TO25:SYSL,0,I,1,40,8:NEXT
220 REM
230 REM FELD 1 =  HAUPTMENUE
240 REM
250 SYSS,1,5,12,18:SYSL,0,1,1,40,8:SYSL,
0,5,1,40,8:SYSL,1,1,12,25,8
260 SYSL,1,1,29,25,8:SYSL,0,2,13,16,10:S
YSL,0,4,13,16,10:SYSL,1,2,13,3,10
270 SYSL,1,2,28,3,10:SYSP,3,16,4,"DEMO -
 PRG"
280 REM
290 REM FELD 2 = SYS......
300 REM
310 SYSS,7,21,4,34:SYSL,0,7,1,40,8:SYSL,
0,21,1,40,8:SYSL,1,1,4,25,8
320 SYSL,1,1,37,25,8
330 N=1:FORI=10TO18STEP4:SYSP,I,6,3,N:N=
N+1:NEXT
340 FORI=9TO19STEP2:SYSL,0,I,6,3,12:NEXT
350 FORI=9TO17STEP4:SYSL,1,1,6,3,12:SYSL
,1,I,8,3,12:NEXT
360 SYSP,10,10,5,"SYS 49152  LINIE  ZEIC
HNEN"
370 SYSP,14,10,14,"SYS 50083  FELD  LOE
SCHEN"
380 SYSP,18,10,13,"SYS 50242  PRINT  BIL
DSCH."
390 REM
400 REM FELD 3 NOCH......
410 REM
420 SYSS,24,24,5,32:SYSL,0,23,1,40,8:SYS
L,0,25,1,40,8
430 SYSL,1,1,4,25,8:SYSL,1,1,37,25,8
440 SYSP,24,11,14,"NOCH  EINMAL  (J/N)"
450 REM
460 REM
470 GETX$:IFX$=""THEN470
480 IFX$="N"THENPRINTCHR$(147):LIST
490 IFX$="J"THEN100
500 GOTO470
READY.
```

Das Programm »Maske« in Form von DATA-Zeilen

```
10 REM-----------------------------------
20 REM UP MASKE  (DATAS)
30 REM    JACQUES EFFENBERG
40 REM-----------------------------------
50 :
1000 DIMP(115):FORI=0TO115:READX:P(I)=X:
SP=SP+X:NEXT
1010 DATA1249,1266,1206,1429,1453,1416,1
184,1949,1754,1732
1020 DATA1665,1376,1854,1202,1653,1611,2
033,1588,1979,1674
1030 DATA1433,1523,1343,1359,1493,1984,1
744,1413,1649,1693
1040 DATA1883,1478,1422,1121,1447,1745,1
952,1318,1747,1954
1050 DATA1519,1860,1592,1701,1429,2066,1
590,1986,1392,2063
1060 DATA1762,1649,1811,1225,1887,1537,1
512,1552,1640,1510
1070 DATA1878,1306,1379,1529,1544,1473,1
799,1798,1244,1667
1080 DATA1282,1244,1703,1404,1342,1319,1
350,1372,1414,1392
1090 DATA1433,1529,1708,2062,1481,1995,1
689,1918,1744,2013
1100 DATA2050,1369,1363,1623,1454,1322,1
282,1264,934,668
1110 DATA808,793,677,826,723,762,716,661
,752,763
1120 DATA736,827,564,788,759,387
1130 IFSP<>168139THENPRINT"FEHLER PRUEFS
UMME !":END
1140 FORI=49152TO50539:READX:POKEI,X:S=S
+X
1150 Z1=Z1+1:S1=S1+X
1160 IFZ1=12ANDS<>P(P1)THEN1210
1170 IFZ1=12THENP1=P1+1:S=0:Z1=0
1180 NEXT
1190 IFS1<>168139THENPRINT"FEHLER LETZTE
 REIHE DATA !":END
1200 PRINT"OK.":END
1210 Z$=STR$(PEEK(64)*256+PEEK(63))
1220 Z1=LEN(Z$)-1:Z2=2:SP=633:PRINTCHR$(
147)
1230 FORI=0TOZ1-1:Z(I)=VAL(MID$(Z$,Z2,1)
):Z2=Z2+1:NEXT
1240 PRINT" ** FEHLER IN ZEILE : "Z$
1250 POKE631,76:POKE632,105
1260 FORI=0TOZ1-1:POKESP,Z(I)+48:SP=SP+1
:NEXT
1270 POKESP+1,13:POKE198,Z1+4
1280 END
1290 DATA 169, 0, 141, 169, 2, 141, 170,
 2, 141, 171, 2, 141
1300 DATA 172, 2, 141, 173, 2, 141, 174,
 2, 141, 175, 2, 141
1310 DATA 176, 2, 32, 43, 193, 224, 2, 1
76, 45, 142, 169, 2
1320 DATA 32, 43, 193, 202, 48, 52, 224,
 25, 176, 48, 134, 252
1330 DATA 32, 43, 193, 202, 48, 56, 224,
 40, 176, 52, 134, 253
1340 DATA 32, 43, 193, 32, 126, 192, 134
, 254, 32, 43, 193, 142
1350 DATA 168, 2, 32, 20, 193, 96, 32, 6
8, 229, 162, 150, 32
1360 DATA 167, 192, 232, 224, 180, 208,
248, 32, 174, 192, 32, 68
1370 DATA 229, 162, 0, 32, 167, 192, 232
, 224, 28, 208, 248, 32
1380 DATA 174, 192, 32, 68, 229, 162, 28
, 32, 167, 192, 232, 224
1390 DATA 56, 208, 248, 32, 174, 192, 32
, 182, 192, 173, 174, 2
1400 DATA 208, 1, 96, 32, 68, 229, 174,
175, 2, 32, 167, 192
1410 DATA 232, 236, 176, 2, 208, 247, 32
, 174, 192, 165, 57, 133
```

Das Programm »Maske« in Form von DATA-Zeilen (Fortsetzung)

```
1420 DATA 20, 165, 58, 133, 21, 32, 19,
166, 32, 201, 166, 189
1430 DATA 153, 196, 32, 210, 255, 96, 16
9, 13, 32, 210, 255, 32
1440 DATA 153, 192, 138, 240, 49, 173, 1
69, 2, 208, 22, 224, 41
1450 DATA 176, 40, 202, 169, 39, 56, 229
, 253, 133, 254, 228, 254
1460 DATA 176, 2, 232, 96, 208, 38, 232,
96, 224, 26, 240, 18
1470 DATA 202, 169, 24, 56, 229, 252, 13
3, 254, 228, 254, 176, 2
1480 DATA 232, 96, 208, 30, 232, 96, 238
, 174, 2, 169, 56, 141
1490 DATA 175, 2, 169, 89, 141, 176, 2,
96, 238, 174, 2, 169
1500 DATA 89, 141, 175, 2, 169, 120, 141
, 176, 2, 96, 238, 174
1510 DATA 2, 169, 120, 141, 175, 2, 169,
150, 141, 176, 2, 96
1520 DATA 173, 169, 2, 201, 1, 240, 4, 3
2, 74, 193, 96, 174
1530 DATA 169, 2, 189, 126, 196, 133, 25
1, 32, 74, 193, 96, 32
1540 DATA 253, 174, 32, 158, 183, 96, 16
6, 252, 32, 240, 233, 165
1550 DATA 253, 240, 11, 165, 209, 24, 10
1, 253, 133, 209, 144, 2
1560 DATA 230, 210, 32, 36, 234, 96, 32,
50, 193, 160, 0, 140
1570 DATA 170, 2, 32, 139, 193, 165, 251
, 145, 209, 173, 168, 2
1580 DATA 145, 243, 173, 169, 2, 201, 1,
240, 9, 200, 140, 170
1590 DATA 2, 196, 254, 208, 22, 96, 238,
173, 2, 198, 254, 240
1600 DATA 13, 230, 252, 32, 50, 193, 32,
139, 193, 173, 169, 2
1610 DATA 16, 211, 96, 32, 139, 193, 173
, 169, 2, 16, 202, 173
1620 DATA 169, 2, 201, 1, 240, 4, 32, 11
2, 194, 96, 32, 38
1630 DATA 195, 96, 173, 169, 2, 208, 36,
173, 171, 2, 208, 14
1640 DATA 198, 252, 32, 50, 193, 32, 231
, 193, 230, 252, 32, 50
1650 DATA 193, 96, 230, 252, 32, 50, 193
, 32, 231, 193, 198, 252
1660 DATA 32, 50, 193, 206, 171, 2, 96,
173, 171, 2, 208, 14
1670 DATA 198, 253, 32, 50, 193, 32, 231
, 193, 230, 253, 32, 50
1680 DATA 193, 96, 230, 253, 32, 50, 193
, 32, 231, 193, 198, 253
1690 DATA 32, 50, 193, 206, 171, 2, 96,
162, 0, 177, 209, 221
1700 DATA 126, 196, 240, 33, 232, 224, 1
1, 208, 246, 173, 169, 2
1710 DATA 208, 11, 162, 0, 189, 126, 196
, 133, 251, 142, 172, 2
1720 DATA 96, 162, 0, 142, 172, 2, 232,
189, 126, 196, 133, 251
1730 DATA 96, 173, 169, 2, 208, 45, 173,
171, 2, 208, 20, 162
1740 DATA 0, 177, 209, 221, 141, 196, 24
0, 214, 232, 224, 4, 208
1750 DATA 246, 162, 1, 142, 172, 2, 96,
162, 0, 177, 209, 221
1760 DATA 137, 196, 240, 194, 232, 224,
4, 208, 246, 162, 1, 142
1770 DATA 172, 2, 96, 173, 171, 2, 208,
20, 162, 0, 177, 209
1780 DATA 221, 149, 196, 240, 180, 232,
224, 4, 208, 246, 162, 1
1790 DATA 142, 172, 2, 96, 162, 0, 177,
209, 221, 145, 196, 240
1800 DATA 160, 232, 224, 4, 208, 246, 16
2, 1, 142, 172, 2, 96
1810 DATA 165, 252, 208, 51, 238, 171, 2
, 32, 154, 193, 173, 172
1820 DATA 2, 208, 1, 96, 173, 170, 2, 20
8, 8, 162, 6, 189
1830 DATA 126, 196, 133, 251, 96, 165, 2
54, 56, 237, 170, 2, 201
1840 DATA 1, 208, 8, 162, 5, 189, 126, 1
96, 133, 251, 96, 162
1850 DATA 8, 189, 126, 196, 133, 251, 96
, 201, 24, 208, 48, 32
1860 DATA 154, 193, 173, 172, 2, 208, 1,
96, 173, 170, 2, 208
1870 DATA 8, 162, 4, 189, 126, 196, 133,
251, 96, 165, 254, 56
1880 DATA 237, 170, 2, 201, 1, 208, 8, 1
62, 10, 189, 126, 196
1890 DATA 133, 251, 96, 162, 7, 189, 126
, 196, 133, 251, 96, 238
1900 DATA 171, 2, 32, 154, 193, 173, 172
, 2, 208, 13, 32, 154
1910 DATA 193, 173, 172, 2, 208, 1, 96,
32, 180, 194, 96, 32
1920 DATA 154, 193, 173, 172, 2, 208, 4,
32, 128, 194, 96, 173
1930 DATA 170, 2, 208, 8, 162, 3, 189, 1
26, 196, 133, 251, 96
1940 DATA 165, 254, 56, 237, 170, 2, 201
, 1, 208, 8, 162, 9
1950 DATA 189, 126, 196, 133, 251, 96, 1
62, 2, 189, 126, 196, 133
1960 DATA 251, 96, 165, 253, 208, 35, 23
8, 171, 2, 32, 154, 193
1970 DATA 173, 172, 2, 208, 1, 96, 173,
173, 2, 208, 4, 32
1980 DATA 133, 194, 96, 165, 254, 201, 1
, 208, 4, 32, 185, 194
1990 DATA 96, 32, 4, 195, 96, 201, 39, 2
08, 32, 32, 154, 193
2000 DATA 173, 172, 2, 208, 1, 96, 173,
173, 2, 208, 4, 32
2010 DATA 151, 194, 96, 165, 254, 201, 1
, 208, 4, 32, 203, 194
2020 DATA 96, 32, 22, 195, 96, 238, 171,
2, 32, 154, 193, 173
2030 DATA 172, 2, 208, 4, 32, 81, 195, 9
6, 32, 154, 193, 173
2040 DATA 172, 2, 208, 4, 32, 54, 195, 9
6, 173, 173, 2, 208
2050 DATA 4, 32, 159, 194, 96, 165, 254,
201, 1, 208, 4, 32
2060 DATA 211, 194, 96, 32, 30, 195, 96,
32, 43, 193, 202, 48
2070 DATA 45, 224, 25, 176, 41, 134, 251
, 32, 43, 193, 202, 48
2080 DATA 33, 224, 25, 176, 29, 134, 252
, 32, 217, 195, 32, 43
2090 DATA 193, 202, 48, 46, 224, 40, 176
, 42, 134, 253, 32, 43
2100 DATA 193, 32, 245, 195, 134, 254, 3
2, 26, 196, 96, 32, 94
```

Das Programm »Maske« in Form von DATA-Zeilen (Schluß)

```
2110 DATA 192, 165, 251, 197, 252, 48, 1
8, 240, 16, 32, 68, 229
2120 DATA 162, 180, 32, 167, 192, 232. 2
24, 211, 208, 248, 32, 174
2130 DATA 192, 96, 32, 110, 192. 138, 24
0, 22, 224, 41, 176, 18
2140 DATA 202, 169, 39, 56. 229, 253, 13
3, 254, 228, 254, 176, 2
2150 DATA 232, 96, 208, 8, 232, 96, 32.
234, 192, 32, 135. 192
2160 DATA 32, 248, 192, 32, 135, 192, 16
6, 251, 32, 240, 233, 165
2170 DATA 253, 240, 11, 165, 209, 24, 10
1, 253, 133, 209, 144, 2
2180 DATA 230, 210, 169, 32, 160. 0. 145
, 209, 200, 196, 254, 208
2190 DATA 249. 228, 252. 240, 4, 230. 25
1, 208, 217, 96, 32, 43
2200 DATA 193, 202, 48, 41, 224. 25, 176
, 37, 134, 214, 32, 43
2210 DATA 193, 202, 48, 32, 224, 40, 176
, 28, 134, 211, 32, 43
2220 DATA 193, 173, 134, 2, 133, 251. 14
2, 134, 2, 32, 253. 174
2230 DATA 32, 119, 196. 165, 251, 141, 1
34, 2, 96. 32, 94, 192
2240 DATA 32, 110, 192, 32, 108, 229. 32
, 164, 170, 96, 64, 93
2250 DATA 91, 107, 109, 110. 112, 113. 1
14, 115, 125, 64, 110. 112
2260 DATA 114, 64. 109, 113, 125, 93. 10
7, 109, 112, 93, 110, 115
2270 DATA 125, 90, 69, 73, 76, 69. 78, 8
7, 69, 82, 84, 32
2280 DATA 60, 49, 32, 79, 68, 69, 82, 32
, 62, 50, 53, 32
2290 DATA 32, 73, 78, 32, 58, 83, 80. 65
, 76, 84, 69, 78
2300 DATA 87, 69, 82, 84, 32, 60, 49, 32
, 79, 68, 69, 82
2310 DATA 32, 62, 52, 48, 32, 73, 78, 32
, 58, 76, 65, 69
2320 DATA 78, 71, 69, 32, 76, 73, 78, 73
, 69, 32, 90, 85
2330 DATA 32, 60, 32, 79, 68, 69, 82, 32
, 90, 85, 32, 62
2340 DATA 32, 32, 73, 78, 32, 58, 83, 80
, 65, 76, 84, 69
2350 DATA 32, 43, 32, 76, 65, 69, 78, 71
, 69, 32, 76, 73
2360 DATA 78, 73, 69, 32, 62, 52, 48, 32
, 32, 73, 78, 32
2370 DATA 58, 90, 69, 73, 76, 69, 32, 43
, 32, 76, 65, 69
2380 DATA 78, 71, 69, 32, 76, 73, 78, 73
, 69, 32, 62, 50
2390 DATA 53, 32, 32, 73, 78, 32, 58, 82
, 73, 67, 72, 84
2400 DATA 85, 78, 71, 32, 87, 69, 82, 84
, 32, 90, 85, 32
2410 DATA 62, 32, 40, 48, 45, 49, 41, 32
, 32, 73, 78, 32
2420 DATA 58, 90, 69, 73, 76, 69, 32, 49
, 32, 73, 83, 84
2430 DATA 32, 62, 32, 65, 76, 83, 32, 90
, 69, 73, 76, 69
2440 DATA 32, 50, 32, 32, 73, 78, 32, 58
READY.
```

Fortsetzung von Seite 13

tiv bald mit allem, was er sieht, unzufrieden sein. Er wird sagen: Verdammt noch mal, ich mache meine eigene Mailbox. Das ist ja das elektronische Äquivalent zu einer Zeitung. Die Medien per DFÜ ermöglichen so etwas für alle, die etwas sagen, etwas mitteilen wollen. Jede Mailbox hat ihre eigene Handschrift, ihre eigenen Informationsecken, wenn auch ziemlich viel soziales Geräusch wie beim CB-Funk dabei ist.

Plant Ihr vielleicht eine Mailbox oder auch etwas ähnliches?

Wir sind dabei, eine elektronische Mitfahrerzentrale zu programmieren. Der Schwarzmarkt hier in Hamburg praktiziert seit acht Jahren eine Mitfahrerzentrale. Wir wollen so etwas über DFÜ und Computer machen.

Kostenlos, oder?

Ja, wenn einer aber etwas spenden will, wird das natürlich gerne gesehen, aber grundsätzlich soll es nichts kosten.

Wie seht Ihr die Lage bei den Modems?

Die Situation bei den Modems sieht so aus: Die billigsten sind im Moment von Tandy, die aber im Moment nicht lieferbar sind. Es gibt noch ganz billige Selbstbaumodems, die aber nur 300 Baud schaffen. Wenn man sich länger damit beschäftigt, sind 300 Baud einfach zu langsam. Da kommt man auf so etwas wie den 79 10/79 11; deshalb haben wir den Bausatz mit Platine entwickelt, den wir zu Selbstkosten vertreiben.

Noch eine Frage: Ist BTX eine Konkurrenz für DFÜ?

BTX ist von der Gebührenpolitik so konzipiert, daß es einfach viel zu teuer ist. BTX wurde zu einem Zeitpunkt entworfen, als die ganzen Mikro- und Kleincomputer noch gar nicht absehbar waren. Noch einmal direkter gesagt: BTX und der Cept-Standard wurden am Schreibtisch konzipiert, und die meisten Bildschirmtextgeräte sind nun einmal reine BTX-Geräte und keine Computer, und zwingen den, der ein BTX-Gerät hat, sich zusätzlich noch einen Computer zu kaufen. So glaube ich einfach, daß ein Datennetz, das die grafischen Möglichkeiten eines C 64 ausnutzt, sich mehr verbreiten wird, als BTX von der Post, das staatlich unterstützt wird.

Also BTX wird längerfristig nicht überleben?

Nicht als breites Enduser-Ding. Für spezialisierte Sachen und bestimmte Firmen wird es wohl überleben, aber in sehr kleinem Rahmen. Guck Dir an wieviel hunderttausend C 64 es gibt und wieviel tausend BTX-Anwender. Trotzdem werden wir bis zum Ende des Jahres unseren BTX-Dienst ausbauen und erweitern, aber ab Einführung der erhöhten Gebühren werden wir unseren BTX ausschalten. Wir haben einen Antrag auf 100 Seiten laufen, um einen einigermaßen aktuellen Informationsdienst anzubieten. Wir wollen eben einige BTX-Anwender auf andere Gedanken bringen und sie auf andere Möglichkeiten hinweisen. Aber spätestens zur Gebührenerhöhung werden wir aussteigen. Wir sind auf der Seite 19058 im Berliner Übergangsrechner, und auf der Seite 20305080 im neuen.

Ok, das war's!

Der CCC ist auch noch im RMI-Net über Datex-P zu erreichen (NUA 44241040341) unter dem Namen Chaos-Computer-Club c/o Schwarzmarkt, Bundesstraße 9, 2000 Hamburg 13.



Wie die Überschrift schon andeutet, hat sich eine Änderung vollzogen. Das »Ka« für Kassette ist weggefallen. Dafür hat sich »Di« zu Disk gemausert. Es ist uns also endlich gelungen, die Programme auch auf Diskette anzubieten. Wir mußten allerdings eine Entscheidung fällen: Kassette oder Diskette, beides ging nicht. Die Diskette ist aufgrund ihrer Verbreitung ausgewählt worden. Dafür sind jetzt alle Programme einer Ausgabe (VC 20 und C 64) auf einer Diskette erhältlich.

Eines hat sich aber nicht geändert: der Preis. Die Diskette für eine Ausgabe kostet demnach 29,90 Mark. Sie werden bei einigen Disketten bestimmte Programme vermissen. Deren Autoren konnten sich nicht entschließen, ihr Programm im Rahmen des Leserservice für eine Verbreitung auf Datenträger freizugeben. Bei den Ausgaben 4, 5 und 6 können noch Kassetten (VC ...) bestellt werden. Auf kurze Programme wurde aus Gründen der Übersichtlichkeit verzichtet. Nun noch einige technische Details. Zu den Programmen sind immer die Seitenzahlen angegeben, unter der Sie die Beschreibungen in der entsprechenden Ausgabe finden können. Der Diskette liegen also keinerlei Informationen bei. Lesen Sie daher aufmerksam die Anleitung (ob SYS-Befehle nötig sind, in welcher Reihenfolge geladen werden muß, eventuelle Sprach- oder Speichererweiterungen und ähnliches mehr) in dem jeweiligen Artikel nach. Aus Aktualitätsgründen wird jeweils die abgedruckte Version angeboten. Eventuelle systematische Fehler, die sich noch im Programm befinden können, müssen von Ihnen selbst, nach Studium des Druckfehlerteufelchens, korrigiert werden.

Ausgabe 10/84

Bestellnummer **CB 019**

Commodore 64
Finanzmathematik (AdM)
Hypra-Load (LdM)
Hardcopy
Compact 2
Hardcopy MPS 801
Hardcopy VC 1526 neu
Hardcopy Gemini-10X
Hardcopy FX-80
Hardcopy VC 1520 farbig
Apocalypse now
Supercopy
Disk-Dump
Diskettenorganisation
User-Port-Tastatur
UP-Maske

VC 20
Epedemic
Video-Vorspann

Ausgabe 9/84

Bestellnummer **CB 014**

Commodore 64
Indexsequentielle Adreßdatei, S. 54 — Spring Vogel (LdM), S. 68 — Orgel/Synthesizer (AdM), S. 70 — Sprite Aid +, S. 89 — Screen Change, S. 94 — List-Stop, S. 97 — Renew, Datawandler, S. 102 — Synthetische suchen, S. 104 — Geregelter Zahlungsverkehr, S. 164

VC 20
Schiebung (GV >), S. 77 — Deuzei (8K >), S. 79 — Hardcopy 1520 (GV >), S. 87 — RS232-Interface (GV >), S. 100 — Datawandler (GV >), S. 102

Ausgabe 8/84

Bestellumner **CB 013**

Commodore 64
Castle of Doom, S. 66 — Pac-Boy, S. 89 — Kopplung, S. 73 — User-Port-Display, S. 97 — RS232-Test, S. 77 — View BAM, S. 99 — Görlitz Hardcopy, S. 83 — Milchvieh, S. 156

VC 20
Kudiplo (3K), S. 86 — Print at Restore n (GV), S. 101

Ausgabe 7/84

Bestellnummer **CB 017**

Commodore 64
Terminalprogramm, S. 24 — Softwarekatalog, S. 72 — Russvok (SB), S. 76 — Crown No. 1, S. 80 — Space Invaders, S. 81 — 1520 Hardcopy, S. 108 — Centronics Interface, S. 110 — Kurvendiskussion, S. 116 — Copy Rel. Files, S. 132 — Autostart, S. 138 — Strubs (OP u. QP), S. 154

VC 20
Rätsel, S. 122

Ausgabe 6/84

Commodore 64
Bestellnummer **CB 018**
Lehrerkalender, S. 64 — Morsetrainer, S. 72 — Supervoc, S. 69 — Grafische Darst. (SB), S. 82 — Hot Wheels, S. 92

VC 20 Bestellnummer **VC 008**
Movemaster (8K), S. 78 — Ghost Manor (GV), S.104 — Logic Disass. (3K >), S. 108 — Underground (LdM 16K), S. 120

Ausgabe 5/84

Commodore 64
Bestellnummer **CB 016**
Adreß- & Telefonregister, S. 64 — Fahrsimulator, S. 82 — Schatzsucher (LdM), S. 90

VC 20 Bestellnummer **VC 007**
Relative Datei (8K), S. 69 — Schmatzer (GV) S. 76 — 3D-Grafik (8K), S. 78 — Rallye (28K), S. 128

Ausgabe 4/84

Commodore 64
Bestellnummer **CB 015**
Drawline + Demos, S. 65 — Sprite Move, S. 70 — Invaders (SB), S. 74 — Caesar, S. 78 — Disk-Copy, S. 92 — Merge, S. 94

VC 20 Bestellnummer **VC 006**
Elektr. Notiz. (16K), S. 50 — Rennfahrer (GV), S. 86 — Erste Hilfe (GV >), S. 88 — Disk-Copy (3K >), S. 92

Bedeutung der Abkürzungen

*LdM = Listing des Monats
*AdM = Anwendung des Monats
*SB = Simons Basic
*GV = Grundversion
*GV > = alle Speicherversionen können verwendet werden (einschließlich GV)
*3K = 3-KByte-Speichererweiterung wird benötigt
*8K > = Speichererweiterung größer als 8 KByte wird benötigt



Fortsetzung von Seite 152

len Addieren. Auch mit dem Übertrag läuft es wie im dezimalen. Beispiel: 2+2=4:

10 entspricht dez. 2
+10 entspricht dez. 2

100, was dezimal eine 4 ergibt.

In der zweiten Spalte wurde nach der Regel verfahren: 1+1=10. Rechnen wir noch 3+3=6:

11 entspricht dez. 3
+11 entspricht dez. 3

110, was dezimal eine 6 ergibt.

In der ersten Spalte wurde gerechnet 1+1=10, wobei nach dem alten Motto: 0 hin, 1 im Sinn die 0 unter den Strich gesetzt wurde. In der zweiten Spalte wird dann so verfahren: 1+1+1 (das ist die 1, die wir »im Sinn« hatten)=11. Ich meine, daß Sie ohne Probleme die folgenden Übungsaufgaben lösen und dann jeweils dezimal das Ergebnis nachprüfen können: 10+5, 7+1, 16+16, 240+16, 62+65.

In der nächsten Folge werden wir eine Anzahl neuer Assembler-Befehle kennenlernen und erfahren, wie der Computer Zahlen voneinander abzieht. Bei der Gelegenheit lernen Sie dann noch einige Flaggen kennen, und wir werden das Rätsel, warum bei LDA # $FF eine negative Zahl angezeigt wird, lösen. Und vor allen, Sie erhalten einen komfortablen Assembler, Disassembler und einen Monitor.

(Heimo Ponnath/aa)

Inserentenverzeichnis

Einem Teil dieser Ausgabe liegen Prospekte der Firmen Interest-Verlag, Kissing und Microcomputerladen, Berlin, bei.

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)
Stellv. Chefredakteur: Michael Scharfenberger (sc)
Redakteure: aa = Albert Absmeier, leitender Redakteur (130), ev = Volker Everts (278), kg = Karin Gößlinghoff (269), gk = Georg Klinge (169), rg = Christian Rogge (278)
Redaktionsassistenz: Dagmar Zednik-Djadja (237)
Fotografie: Janos Feitser, Titelfoto: Alex Kempkens

Layout: Leo Eder (Ltg.), Dagmar Berninger, Willi Gründl, Walter Höß, Cornelia Weber

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug, Tel. 042-22 31 55/56, Telex: 862329 mut ch
USA: M&T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. 001-4240600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180), Leo Eder (181)

Anzeigenleitung: Peter Schrödel (156)
Anzeigenverkauf: Alfred Reeb (211)

Anzeigenverwaltung und Disposition: Michaela Hörl (171)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 1 vom 1. März 1984.
Anzeigengrundpreise: 1/1 Seite sw: DM 7400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/1-Seite
Anzeigen im Einkaufs-Magazin: Die ermäßigten Preise im Einkaufs-Magazin gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 5400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 10,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Plieninger Straße 100, 7000 Stuttgart 80 (Möhringen), Telefon (07 11) 72004-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-1 19. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,-. Der Abonnementspreis beträgt im Inland DM 72,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: Druckerei E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly.
Für Anzeigen: Peter Schrödel.

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.